

# AMIGA DOS

5/90

ISSN 0937-2717  
DMV-Verlag

## Viele Tips und Tricks

- Linkchecker
- Modula-2
- CLI und BASIC für Einsteiger

## DTP

Schweres Gerät  
im Praxistest

Music Power wie noch nie

## TFMX Soundeditor

## Profi Genlock

Das Studio zu Hause

## Top-Listings

Superschneller  
Fraktalgenerator

Komfortabler Filemonitor

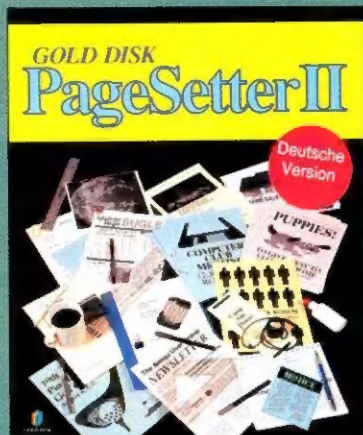
## Test

Deluxe Video

Citizen Swift 24

MIT  
GROSSEM  
SPIELE-  
TEIL

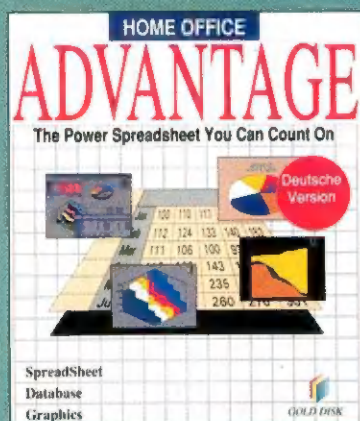




PAGESETTER II - das heißt professionelles Desktop-Publishing zum Low Cost-Preis.

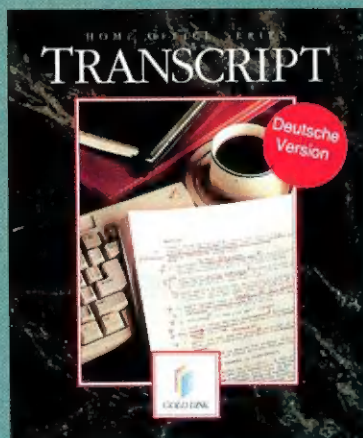
Durch die Verwendung der AGFA Compugrafik-Vektorschriften garantiert PageSetter II für eine professionelle Ausgabequalität in höchstmöglicher Auflösung auf allen grafikfähigen Preferences-Druckern. Neben den im Programm integrierten Grafikfunktionen, können sie sowohl IFF- als auch Vektorgrafiken direkt in Ihre PageSetter II-Dokumente einbinden.

Der Preis macht die Sensation perfekt: deutsche Version 198,-



THE ADVANTAGE - klipp und klar.

The Advantage ist eine professionelle Tabellenkalkulation mit außergewöhnlich großen Arbeitsblättern (65.000 Zeilen x 65.000 Spalten), ARexx-Unterstützung, integrierten umfangreichen Datenbankfunktionen, Makros, 2- und 3-dimensionale Grafikauswertung, Grafiken in IFF- und Vektorformat speicherbar (DTP), umfangreiche mathematische, finanzmathematische und statistische Funktionen, automatische Konvertierung von 1-2-3 Datenblättern und und und ... deutsche Version DM 248,-



TRANSCRIPT - Vorsicht! Diese Textverarbeitung entspricht nicht dem Standard.

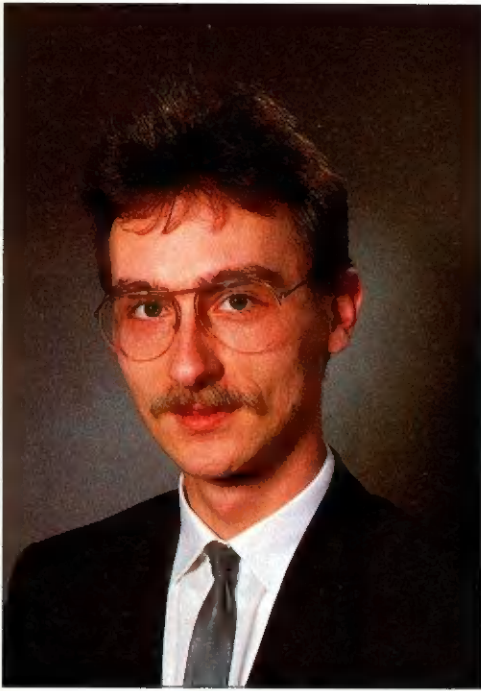
Transcript ist eine Textverarbeitung und kein Versuch zu einem DTP-Programm. Deshalb überläßt Transcript das Einbinden von Grafiken und nicht druckereigenen Schriften professionellen DTP-Programmen wie PageSetter II. Transcript ermöglicht eine effiziente Textbearbeitung: Textgröße nur vom Speicher abhängig, Makros, Preview-Modus, Serienbriefe, Floskeltasten, mehrere Texte gleichzeitig, PageSetter II und Professional Page Online-Support, automatische Indexerstellung, sehr hohe Arbeitsgeschwindigkeit und das alles zu einem Preis von DM 99,- für die deutsche Version.



**GOLD DISK**

Marktplatz 16 • 4018 Langenfeld  
Tel.: 02173 - 71093 • Fax: 02173 - 82799





# Desktop-Video im Aufwind

**D**ie Zeiten, in denen die Erstellung professioneller Videos Fernsehstudios mit ihrem millionenschweren Equipment vorbehalten waren, sind vorbei. Der Amiga hat längst Einzug auch in diese Studios gehalten und vermag Gerätschaften zu ersetzen, die noch vor einigen Jahren den durchschnittlichen Fünf-Jahres-Lohn eines Angestellten verschlungen hätten. Dies bedeutet für den Heimanwender, daß auch zu Hause jetzt Projekte in Angriff genommen werden können, deren Realisierung vor einigen Jahren schlichtweg unmöglich gewesen wäre. Die Preise für benötigtes Zubehör wie Genlocks, RGB-Splitter und Digitizer, aber auch für entsprechende Software haben einen Stand erreicht, der jedem die Chance gibt, für wenig Geld seine Ideen in die Tat umzusetzen. Bei allem Enthusiasmus über diese Möglichkeiten sollte man allerdings den Zeitaufwand nicht außer acht lassen. Wer sich über kleinere, einfach zu realisierende Projekte zunächst Sicherheit im Umgang mit der Materie verschafft, kann viele entstehende Probleme schon im Vorfeld klären und hat so wahrscheinlich größere

Aussichten, auch spektakuläre Videos zu erstellen. Apropos spektakulär: Besonders möchten wir Sie auf den Fraktalgenerator in dieser Ausgabe hinweisen, der nicht nur durch seine Farbenpracht beeindruckt. Außerdem ist er der zur Zeit schnellste Fraktalgenerator, der bis dato in einer Zeitschrift für den Amiga veröffentlicht wurde. Auf vielfachen Wunsch haben wir auch noch einmal die Problematik der Linkviren unter die Lupe genommen.

Unsere Tips-und-Tricks-Ecke bietet Ihnen eine wirksame Waffe gegen die ungeliebten Eindringlinge, denen schon manche Diskette zum Opfer gefallen ist. Selbstverständlich hat AMIGA DOS noch wesentlich mehr zu bieten, egal ob Sie gerade erst einsteigen oder sich schon länger mit dem Amiga beschäftigen. Doch überzeugen Sie sich selbst. Sicherlich können wir nicht alle Fragen im Rahmen unseres Heftes klären, Hilfe ist jedoch nur einen Telefonanruf weit entfernt. Bleibt uns nur noch, Ihnen viel Spaß bei der Lektüre der neuen AMIGA DOS zu wünschen, denn uns hat es viel Freude gemacht, dieses Heft für Sie zusammenzustellen.

Herzlichst Ihr

Markus Matejka

Markus Matejka



## NEWS & TRENDS

- Rund um den Amiga 6
- Schlüsseltechnologie 10

## SOFTWARE

- Deluxe Video III 12
- Videoproduktion zu Hause
- Schönschrift für den Matrixdrucker 14
- Postscript jetzt auch auf Nadeldruckern

## HARDWARE

- Mäusemächtige Eingabemedien 17
- Desktop Publishing professionell 18
- Commodores Profianlage im Test
- Merkens MaxiGen 24
- Profi-Genlock mit Spitzenqualität
- TFMX – Ein Soundgigant? 26
- Music Power für den Amiga
- Überzeugendes Mathepaket 28
- Math Treasures-2.0
- Basteleien nicht nur für Anfänger 33
- Resetschalter im Eigenbau
- Auf die Plätze, fertig, Druck 34
- Citizen Swift 24

## TIPS & TRICKS

- Sendung läuft 30
- Datenübertragung leichtgemacht
- LinkChecker V2.0 36
- Kein Platz für Linkviren
- LinkList 40
- Das Listing zum LinkChecker
- Dem CLI Beine machen 44
- Meister Klecksel 76
- Escape-Steuersequenzen in Assembler
- Gewußt wie 88
- Short-Tips rund um den Amiga für Einsteiger und Fortgeschrittene
- Ruft die Mäuse auf den Plan 92
- Printer Control Window in C

## KURS

- Aufsteigerwissen Assembler 96
- Assembler für Profis
- Amiga-BASIC 104
- Programmieren mit Erfolg



Bild 1. Logikspiele erfreuen sich großer Beliebtheit. Quadrant 500 kann Ihnen da so manche harte Nuß zum Knacken aufgeben.

Seite 73

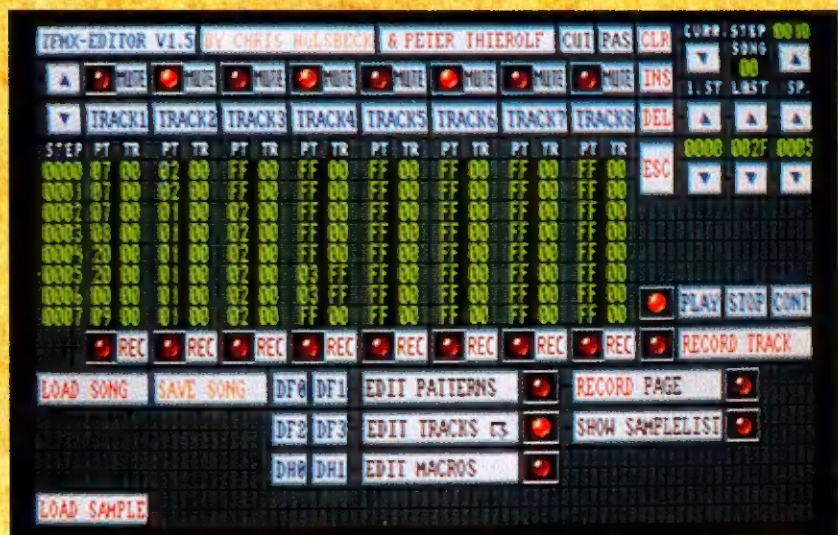


Bild 2. TFMX – ein Soundeditor, der mit vielen Möglichkeiten aufwarten kann. Was TFMX leisten kann, lesen Sie auf

Seite 26



Bild 3. Hoyle Official Book of Games beinhaltet einige der beliebtesten Kartenspiele der Welt

Seite 142



# AMIGA DOS



Bild 4. Die DTP-Anlage von Commodore ist in Verbindung mit entsprechender Software eine DTP-Alternative. Lesen Sie den Praxisbericht zur Arbeit mit DTP auf

Seite 18



Bild 5. Begleiten Sie uns in die faszinierende Welt der Fraktale – mit unserem Super-Listing Fraktalgenerator, dem schnellsten Fraktalgenerator, der je in einer Zeitschrift veröffentlicht wurde

Seite 80

## LISTING

Wie sieht's denn da aus? <i>Filemonitor in Assembler</i>	48
Quadrant 500 <i>Logikspiel in Amiga-BASIC</i>	73
Faszination Fraktal <i>Der derzeit schnellste Fraktalgenerator</i>	80
SprEd <i>Sprite-Editor mit 16 Farben (Teil 3)</i>	111

## WERKSTATT

Arbeiten wie die Profis mit dPaint III (Teil 5)	68
Modula-2 <i>Tips &amp; Tricks in M2 Amiga</i>	116

## PUBLIC DOMAIN

Public-Domain-Werkzeugkiste	122
Public-Domain-Spiele	126

## SPIELE

Spieletests	
5th Gear	128
Bad Company	128
Aquanaut	129
Bodo Illgner Super Soccer	129
Dungeon Quest	136
John Lowes Ultimate Darts	136
Star Trash	137
Hawaiian Odyssey	137
The Cycles	138
Footballer of the Year 2	138
Stryx	140
Spy vs. Spy	141
After the War	141
Hoyle Official Book of Games	142
Fire	142
Space Harrier II	143
Lords of War	143
AMIGA-DOS-Spieletips <i>Helpline</i>	130
Demnächst auf Ihrem Computer <i>Preview</i>	144

## RUBRIKEN

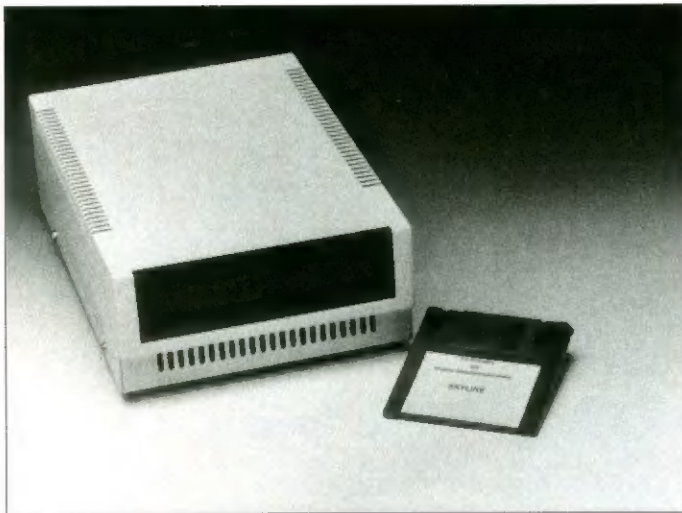
Editorial	3
Leserbriefe	62
Tüftelecke	66
Wettbewerbe	115, 139
Bücher	118
Impressum	145
Inserentenverzeichnis	145
Vorschau	146



### Skyline-Autoboot-Festplatte für A500/1000

Die Firma Skyline bietet eine neue Autoboot-Festplatte für den Amiga 500 und den Amiga 1000 mit neuer grafischer Oberfläche sowie benutzerfreundlicher Menüführung an. Diese Festplattenlösung beinhaltet ein 5,25-Zoll-Hard-

disklaufwerk, Kabel, Klein- teile sowie entsprechende Software zum Autoboot. Vier unterschiedliche Größen von 20, 30, 40 und 60 MByte sind durchgängig mit einem Omti-Controller versehen.



Skyline Autoboot-Festplatte

### Neues von NORDIC POWER

NORDIC POWER, das Freeze-Modul (getestet in der AMIGA DOS 04/90) besitzt seit März 1990 einige neue Eigenschaften. Hier sind sie: 1.) Eine eingebaute Dia-Show ermöglicht es, jede Art von IFF-Grafiken, die mit NORDIC POWER 'eingefroren' und abgespeichert wurden, als Komplett-Vorführung wiederzugeben. Dabei kann es sich auch um Interlace-Bilder handeln. Die Bilder müssen dazu nur auf im Amiga-Format formatierten Disketten vorliegen. Die Reihenfolge der Bildwiedergabe, den automatischen Ablauf und die manuelle Bilderwahl kann man einstellen.

2.) Der Cheat-Modus ist ebenfalls neu, besser bekannt als Poke-Finder. Es ist jetzt also möglich, im Speicher des Amiga automatisch nach Werten zu suchen, die die Anzahl der Leben repräsentie-

ren. Ebenfalls ist es möglich, jetzt Trainer-Pokes aus Zeitschriften ohne Probleme einzugeben und zu nutzen.

3.) Ein Compacker befindet sich jetzt ebenfalls im Modul, wobei die Möglichkeit besteht, zwischen den verschiedenen RAM-Größen und 'PACK' zu wählen. Allerdings sollte sich außer dem 'gefrorenen' Spiel nichts anderes im Speicher befinden. Die vor dem März gekauften NORDIC der S-Serie (SC oder SP) können gegen einen Aufpreis von 29,- DM zuzüglich 10,- DM Versandkosten aufgerüstet werden. Im Lieferumfang sind eine Einbauanleitung für das Update und die Anleitungsergänzung enthalten.

**Info: Data & Electronics  
Venlo B.V.  
Postbus 3119  
NL-5902 RC Venlo  
Niederlande**

### Version 5.0

Neuigkeiten sind auch von der Firma Manx Software Systems Inc. zu vermelden. Seit

Beginn dieses Jahres liefert Manx die neueste Version ihres Aztec C-Compilers aus.

### 50-MHz-Board von Intelligent Memory

Für den Amiga 2000 hat die Firma Intelligent Memory ein neues Board entwickelt: Hurricane 2800 MK II-50.

Eine erste Vorstellung dieses Produkts fand am 9.2.1990 in Paris statt. Anhand eines Raytracing-Bildes, das in einer fantastisch hohen Geschwindigkeit aufgebaut und umgesetzt wurde, konnte man die Leistungsfähigkeit dieses Boards bereits erahnen. Das Topmodell Hurricane 2800 MK II-50 wird demnächst mit einer MC 68030 RC-50 CPU erhältlich sein. Dieses Board läuft mit einer Frequenz von 50 MHz und erreicht eine relative Geschwindigkeit, die rund 20mal schneller als der Standard-Amiga ist.

Der Preis dieses Produkts wird bei zirka 4500 DM inklusive CPU und weiteren Unterlagen liegen. Es kann mit bis zu 16 MByte dynamischen RAMs aufgerüstet werden. Das Board ermöglicht auch ein schnelles SCSI-Interface, das zur Steuerung jeglicher Standard-SCSI-Festplatten verwendet werden kann, und ist voll autobootfähig unter

Kickstart 1.3. Ein Performance-Test mit dem SCSI-Port am Hurricane 2800 MK II-50 ergab Schreib-/Lese-Zugriffsgeschwindigkeiten bei 550 kByte bis zu 500 Directory-Scans pro Sekunde und rund 200 Create/Close pro Sekunde, bezogen auf das Standard-Festplatten-Testprogramm "DPERF2". Der Ronin-Sieve-Benchmark-Test, der in der Amiga World 4'90 veröffentlicht wurde, ergab im Vergleich zu einem ähnlichen 68030-Board folgende Werte: Ein Ablauf mit einer ständigen Frequenz von 33 MHz lief beim "normalen" 68030-Board in 4,25 Sekunden ab, wohingegen das Hurricane 2800 MK II-50 genau 3,00 Sekunden benötigte. Weitere Beispieldaten zu diversen Tests:

- Zum Zeichnen eines Kreises (100mal) wurden 2,00 Sekunden benötigt.

- Sys-Partition-Speedtest lief bei 2500 Iterationen rund 0,84 Sekunden.

Wir werden Sie über weitere Einzelheiten in einer der nächsten Ausgaben näher informieren.

### Eine umfassende Übersicht

...über aktuelle Fachliteratur und Software- bzw. Hardware-Produkte rund um den Amiga hat der Technik Support Verlag im Auftrag der Firma Commodore herausgegeben. Der Amiga Katalog 90 umfaßt rund 320 Seiten, kostet 20,- DM und ist ab Mitte März in Warenhäusern, im Fachhandel und im Buchhan-

del erhältlich. Hier steht nicht die Quantität, sondern die Qualität der Produkte im Vordergrund. Die Palette reicht von neuesten Publikationen zum Amiga über brandaktuelle Themen wie DTP oder Desktop Video bis hin zu den neuesten Entwicklungen im Hardwarebereich wie Turbo Boards etc.

### Termine\*\*\*Termine\*\*\*Termine

#### ● BaCom

Zum dritten Mal findet vom 19. bis zum 21. April im Baden-Badener Kongreßzentrum die südwestdeutsche Büro- und Computerfachmesse statt. Besondere Beachtung findet das Rahmenprogramm "Digital Design", wo neueste Präsentationen aus der Welt der Computergrafik und -animation vorgestellt werden.

#### ● MECOM

In Saarbrücken findet in der

Zeit vom 18. bis zum 20. Mai die MECOM statt.

#### ● BASCOM

Basel ist Treffpunkt der Computer-Freaks in der Zeit vom 09. bis 13. Mai.

#### ● Summer CES

Die internationale Summer Consumer Electronics Show, kurz CES genannt, findet in der Zeit vom 2. bis zum 5. Juni in Chicago statt.



## 'En Passant' – die Schach-Datenbank

Für Freunde des königlichen Spiels ist es oft wichtig, sich auf anstehende Spielbegegnungen (Turniere, Meisterschaften oder auch nur zu Hause) vorzubereiten. Aber es ist oft auch zeitraubend, wenn man erst eine Menge Bücher durchsehen muß, bis man an bestimmte Spielzüge oder Spielbegegnungen kommt oder sie schlimmstenfalls gar nicht erst findet.

Von der Firma Habermehl und Sander gibt es zu diesem Zweck ein neues Amiga-Programm namens 'En Passant'. Hierbei handelt es sich um eine Schach-Datenbank, in welcher der Schachspieler eine Vielzahl an Partien, Zügen und Wissen findet. Bei den Partien kann eine Analyse herangezogen werden, in der es möglich ist, eigene

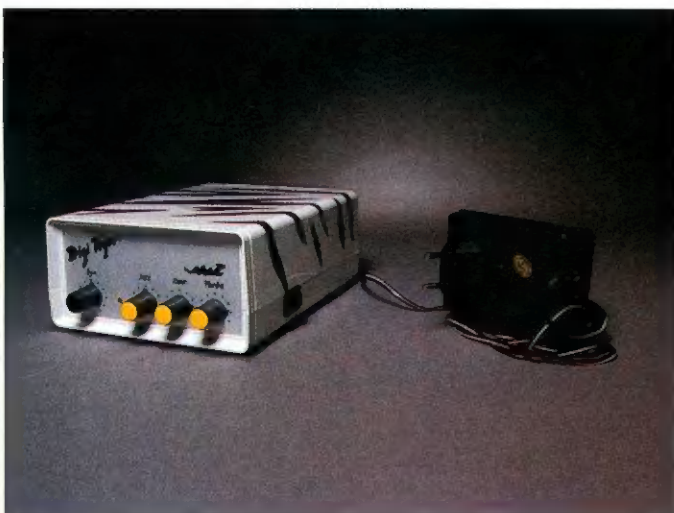
Züge der Probe halber einzubringen und das Ergebnis zu sehen. Sind die eigenen Züge nicht erfolgreich, kann der Spieler zur Ausgangsstellung zurückgehen und einen neuen Versuch starten.

Das Programm läuft auf allen Amiga mit mindestens 1 MByte Speicher und einem Laufwerk. Festplatten werden vom Programm unterstützt, so daß die Datenbank schnell bereit ist. Im Lieferumfang sind Programmdiskette, Datendiskette mit über 600 Großmeisterpartien sowie ein ausführliches deutsches Handbuch enthalten.

**Info:**  
**Habermehl & Sander**  
**Tannenweg 11**  
**3575 Kirchhain 1**  
**Tel.: 06422/7222**



'En Passant' – die Schach-Datenbank



DigiTiger im Blickpunkt

## Westernheld am Bildschirm



Der Westphaser – Schießen elektronisch

'Westphaser' – so nennt sich eine Lichtpistole von Loricel, mit der man auf den Bildschirm schießen kann, ohne Bruchstücke zu hinterlassen. Im Westphaser ist ein lichtempfindlicher Halbleiter-Baustein eingebaut, der Lichtreflexe vom Bildschirm in elektrische Signale umwandelt und diese über die serielle Schnittstelle an den Amiga übergibt. Dem Paket liegt ein Action-Schießspiel bei, das eine fröhliche Balle-

rei im Wilden Westen zum Inhalt hat. Auf der Diskette befindet sich ebenfalls ein Beispielspielprogramm in Amiga-BASIC zum Einlesen der Informationen über die Schnittstelle. Weitere Spiele für den Westphaser sollen bald erscheinen. Zu beziehen ist der Pseudo-Revolver bei

**Rushware**  
**Bruchweg 128**  
**D-4044 Kaarst**  
**Tel.: 02101 / 6070**

## DigiTiger – Video Digitizer V1.0

Die Firma Klaus-D. Tute aus Hannover hat ein neuartiges Video-Digitizer-System auf den Markt gebracht. Mit dem eingebauten RGB-Splitter und dem SP+-Bildspeichersystem ist es möglich, qualitativ hochwertige Digitalisierungen von Schwarzweißbildern und Farbvorlagen von Videoquellen relativ schnell und auch problemlos zu erstellen. Die Bedienung erfolgt über Menüs oder Tastatureingaben und ist durch die deutsche Programmführung vollkommen unproblematisch. Das Speichern und Laden der Bilder erfolgt im Standard-IFF-Format wahlweise mit Icon. Mit einer zusätzlichen Umschaltbox (die leider nicht im Lieferumfang enthalten ist) können die Bilder sofort ausgedruckt werden. Das Programm bietet eine Auswahl zwölf verschiedener Bildauflösungen einschließlich Overscan. HAM-Bilder kön-

nen mit 4096 Farben in 20 bis 45 Sekunden übertragen und speicherfertig berechnet werden. Die Digitalisierungszeiten liegen bei rund 2 Sekunden pro Bild in Schwarzweiß und 20 Sekunden in Farbe bei freier Einstellung von Helligkeit, Kontrast, Farbsättigung, Farbton und Schärfe. Das Programm ist auf allen Amiga-Geräten mit 10 von 12 Auflösungen schon ab 512 kByte Speicher (HAM-Interlace ab 1 MByte) voll lauffähig.

Eine Demodiskette ist für 10,- DM bei der Firma Tute erhältlich und ermöglicht, alle Funktionen zu testen. Der DigiTiger kostet einschließlich Software, Kabel, Netzteil und 50seitigem deutschem Handbuch 698,- DM.

**Klaus D. Tute**  
**Mathildenstr. 12**  
**3000 Hannover 91**  
**Tel.: 0511/629825**



## Im Blickpunkt: GREMLIN

Die Firma Gremlin Graphics Software Limited hat einige neue Spiele angekündigt.

### RAMROD

Ein Spiel, bei dem man die Wahl zwischen zwei recht unterschiedlichen Charakteren hat: Auf der einen Seite Ramrod, der muskelbepackte, egoistische Charakter, dessen Name für den Titel Pate stand. Auf der anderen Seite Rambot, ein nett anzusehender blau-metallisch-glänzender Roboter. Danach steigt man in das Geschehen ein, genauer gesagt in ein dreidimensionales Terrain mit vielen Klippen, Kanälen und Hügeln. Dort kann man dann mit seinem gewählten Charakter auf Punkte- und Münzjagd gehen. Stellt man sich auf kleine glänzende Scheiben, kann man zu fast jedem beliebigen Ort teleportiert werden. Mit dem Fahrstuhl gelangt man direkt in den Arcadenteil – einem Spiel im Spiel – wo man sich in vier unterschiedlichen Disziplinen pro Level austoben kann. Jedes einzelne muß erfolgreich beendet werden, um ins nächste Level zu gelangen. Einige Stellen sind sehr schwierig zu erreichen, hier helfen dann sogenannte Luftlöcher weiter. Fällt man in eines hinein, dämpfen sie den Fall. Steigt man auf eines auf, können sie einen emporheben.

Eines ist besonders wichtig: Ramrod oder auch Rambot müssen ständig in Bewegung gehalten werden, da sonst Lebensenergie verlorengeht. Es gibt eine Vielzahl unterschiedlicher Bonusarten: "Speed", "Bopper", "Super 8" oder "8 way" – rundum alles, was ein Held wie Ramrod zum Bestehen seiner Abenteuer benötigt.

Die vier Levels sind sehr abwechslungsreich gestaltet und fördern damit gleichzeitig den Spielspaß. Level 1 ist das High-Tech-Level, wo die Oberfläche durch Computererstellte Bilder auffällt. In die Zeit der industriellen Revolution fühlt man sich bei Level 2 – dem Low-Tech-Level zurückversetzt. Level 3 ist das Aztec-Level, wo unter anderem der Tempel des Quetzalcoatl und dessen heiliger Altar zu finden sind. Anfang und Ende bildet das Fantasy-Level, auch Fantastec genannt. Kurzum ein interes-

santes Spiel im 3D-Format, das für den Amiga ab März 1990 im Handel erhältlich ist.

### Combo Racer

Eine Simulation für Rennen von Motorrädern mit Seitenwagen, mit optionalem Ein- oder Zwei-Spieler-Modus. Spielt man mit zwei Spielern, übernimmt der zweite die Kontrolle des Beifahrers. Das Spiel verfügt über acht unterschiedliche Rennschauplätze mit den verschiedensten Szenarien. Daneben ermöglicht ein spezielles Feature, der Track-Editor, sich seinen eigenen Rennkurs zu gestalten. Ein Spiel für Rennsportfreunde, das 3D-Routinen enthält und für den Amiga ab April im Handel zu erwerben ist.

### Skidz

In den Straßen der Großstadt ist man mit einem BMX-Rad oder dem Skateboard unterwegs. Das Ziel: der schärfste Typ in der Stadt zu sein. Insgesamt sind sieben Levels in sieben Tagen zu bewältigen. Dabei ist man auf den Straßen, den Parks, im berühmten Chinatown und an anderen Schauplätzen unterwegs. Ein zusätzlicher Arcadenteil vervollständigt das Spiel.

Jeder Parcours muß erfolgreich beendet werden, um zum nächsten Level zu gelangen. Jeder Ort verfügt über eine Vielzahl von Gefahren, die zu überwinden sind. Es ist entscheidend, wie man in der Zeit, die man zur Verfügung hat, lernt, mit dem Rad oder dem Skateboard umzugehen. Am besten ist, man verdient etwas Geld, in dem man Verbrecher fängt, und bessert mit diesem Geld seine Ausrüstung im Laden auf. Mit einer perfekten Ausstattung kann man dann die verschiedensten "Stunts" ausprobieren, bevor man sich ins Gewühl stürzt. Denn dort hat man es mit Räubern, Katzen und Hunden, radfahrenden Tramps oder auch Tauben zu tun. Skidz, ein Spiel, das neugierig macht, ist ab März für den Amiga erhältlich.

### Ultimate Golf und Shark Attack

Ultimate Golf ist eine realistische Golfsimulation, die auf Originalschauplätzen und realen Informationen basiert. Hier sind alle Features eines natürlichen Golfplatzes wie

beispielsweise diverse Bunker, Bäume, Sträucher und Hügel so weit es geht naturgetreu auf einem dreidimensionalen Terrain umgesetzt worden. Interessant sind digitalisierte Sequenzen einzelner Schläge des bekannten Golfers Greg Norman, einer Koryphäe auf diesem Gebiet. Daneben besteht die Möglichkeit, auf den einzelnen Kursen umherzugehen. Ein computerisierter Caddy hilft mit seinen Ratschlägen, die Wahl des Schlägers zu entscheiden. Insgesamt siebzehn verschiedene Eisen stehen dem geübten Golfer zur Verfügung, dazu noch jede Menge unterschiedlicher Effekte. Besonders zu erwähnen: Ultimate Golf und Shark Attack sind im Doppelpack zu bekommen, das heißt, zwei Spiele sind zum Preis von einem und ab sofort für den Amiga zu einem Preis von £ 24.99 zu erhalten.

### IMPOSSAMOLE

Monty, der Maulwurf, ist zum x-ten Male seit 1987 wieder unterwegs.

Er liegt unter einer Palme am Strand und harret der Dinge, die da kommen. Doch plötzlich ... fällt ihm eine Kokosnuß auf den Kopf. Er läuft ziellos umher, ohne den merkwürdigen Schatten zu bemerken, der über ihm schwebt. Ein greller Blitz, und Monty findet sich in Gestalt zweier Schattenfiguren wieder. Die geheimnisvolle Umwandlung kann beginnen. Monty ist urplötzlich Impossamole, der allgegenwärtige, weise Maulwurf der 90 Jahre, dessen Zuhause die Straße ist.

So beginnt ein neues Abenteuer, das unser kleiner Freund Monty bestehen muß. Folgen Sie Monty auf seinem Weg durch das Labyrinth von Tunneln und Höhlen in den insgesamt vier unterschiedlichen Levels. Drei Arten von Waffen stehen Monty zur Verfügung: Laser, Bomben und Kugeln. Begleiten Sie ihn bei seinem gefährvollen Abenteuer, es dürfte sicherlich interessant werden. Impossamole wird ab April für den Amiga im Handel erhältlich sein.

### Federation Quest One – BSS Jane Seymore

Vor zehn Jahren verließ das große Forschungsraumschiff

Jane Seymore mit einer Mission die Erde. In den Tiefen des Orion-Arms der Galaxie sollten unbekannte Welten nach Lebensformen erforscht werden. Ein falsch berechneter Hypersprung brachte dann die Katastrophe. Der Wiedereintritt erfolgte zu nahe einem Neutronenstern, so daß das Schiff in eine tödliche Strahlung geriet. Der größte Teil der Crew starb oder wurde so krank, daß sie das Gedächtnis verloren und keine Ahnung mehr hatten, welche Aufgabe sie an Bord zu bewältigen hatten. Das Überlebenssystem des Schiffes wurde bei dem defekten Hypersprung schwer beschädigt, so daß die Überlebenden kaum eine Chance hatten, dies zu reparieren, um nach Hause zurückkehren zu können. Viele Lebensformen konnten entkommen und besetzten das Schiff. Ein Funkpruch konnte jedoch noch abgesetzt werden.

Viele Lichtjahre davon entfernt im Hauptquartier der Föderation wurde dieser Funkpruch aufgenommen und eine Crew zur Rettung des havarierten Schiffes gebildet.

Ihre Aufgabe ist es, als Kommandant des Rettungsschiffes, die BSS Jane Seymour zu orten, zu reparieren, die entlaufenen Lebensformen zu fangen und das Schiff wieder nach Hause zu bringen. Das Spiel vereinigt eine Menge unterschiedlicher Features von Strategie bis Rollenspiel in sich. Die Szenarien erscheinen jeweils im dreidimensionalen Blickwinkel. Eine Statusanzeige gibt Auskunft über Gesundheit und Umgebung des Spielers. Über 30 verschiedene Alienformen können Ihnen die Mission ebenso erschweren wie defekte Roboter oder die "verrückte" Crew an Bord. Das Spiel erstreckt sich über zwanzig verschiedene Levels, die jeweils über ein Passwort zu erreichen sind. Daneben steht dem Spieler und auch seinen Gegnern ein großes Arsenal an unterschiedlichen Waffen zur Verfügung. Man bewegt sich in über 100 Räumen auf drei verschiedenen Ebenen des Schiffes. Ein Spiel für lange anhaltenden Spielspaß. Federation Quest One für den Amiga ist ab März im Handel erhältlich.



## Amiga-Grafik hochwertig ausgedruckt



Amiga-Bilder hochwertig ausgedruckt...

So schön man mit dem Amiga zeichnen und malen kann, so schlecht sieht das ausgedruckte Ergebnis meistens aus. Querstreifen ruinieren jedes noch so sorgfältig erstellte Bild.

Eine Abhilfe bietet die Firma CGD Dr. Buddemeier an. Wer keinen Farbdrucker mit hohen Qualitäten besitzt oder die entworfenen Grafiken professionell nutzen möchte, kann die Vorlagen an die Firma CGD senden. Diese wer-

den mit hochwertigen Tintenstrahl- und Thermotransferdruckern streifenfrei auf Papier gebracht. Eine Größe bis DIN A4 ist dabei möglich, bei Tintenstrahldruck auch größer.

Kostenlose Infos mit Druckbeispielen können angefordert werden bei

**CGD Dr. Buddemeier**  
Schlesienstr. 40  
D-4400 Münster  
Tel.: 0251 / 62214



... mit Tintenstrahl- und Thermotransfer-Druckern

## Mehr Speicher für den A 500

'ERAM MEGA' und 'Megamodul' heißt eine RAM-Kombination für den Amiga 500, die diesen mit einem zusätzlichen Speicher bis zu 2,3 MByte aufrüstet. Das Besondere daran ist, daß der zusätzliche Speicher nach und nach eingebaut werden kann. 'ERAM MEGA' bildet dabei die Grundstufe. Auf der Platine

befinden sich neben einer akkugesperrten Uhr 512 kByte RAM in Form von MBit-Chips. Als nächstes kann dann das 'MEGAMODUL' zusätzlich eingebaut werden; es enthält in der Grundversion noch einmal 512 kByte. Dieses Modul läßt sich nun nacheinander in der folgenden Konfiguration mit MegaBit-

Chips aufrüsten: 1,5 oder 2 oder 2,3 MByte. Beide Platinen verfügen über vergoldete Anschlußstecker, sind abschaltbar und werden mit deutscher Einbauanleitung ausgeliefert. Die Preise für 'ERAM MEGA' und 'MEGAMODUL' sehen so aus:  
ERAM MEGA 199,- DM,

MEGAMODUL 250,- DM (Grundversion), 375,- DM (auf 2 MByte, nur mit Kickstart 1.3), 489,- DM (auf 2,3 MByte).

**Info: Tröps + Hierl**  
Computertechnik GmbH  
Jordanstr. 3  
D-5040 Brühl  
Tel.: 02232 / 45018

## 'Gefrorene' Programme

Das 'Amiga Action Replay' ist ein neues Freezer-Modul, das über eine Vielzahl von Möglichkeiten verfügt. Der Speicher des Amiga kann ausgelesen, enthaltene Programme können in gepackter Form auf Diskette abgespeichert werden. Das Modul erzeugt auf der Diskette ein eigenes Format, auf dem bis zu drei Programme abgespeichert werden können. Die Programme können hinterher auch ohne Modul wieder geladen werden.

Ein Trainer sucht in Spielen die Werte für 'Leben' heraus und ermöglicht eine Änderung. Der enthaltene Sprite-Editor bringt Sprites aus Programmen auf dem Bildschirm, wo sie geändert werden kön-

nen. Zum Schutz vor Viren enthält das Modul einen Virusdetektor, mit dem die bekanntesten Viren erkannt werden können. Bilder und Musik können separat abgespeichert werden, eine Zeitlupenfunktion ermöglicht das langsamere Ablaufen von Programmen.

Ein 68000-Assembler-Disassembler ermöglicht das Einsehen in Quelltexte und das Editieren von Quelltexten.

Das 'Amiga Action Replay' kostet 189,- DM zuzüglich Versandkosten und ist zu beziehen bei

**Eurosystems**  
Hühnerstr. 11  
D-4240 Emmerich  
Tel.: 02822/45589

## Samurais Edelwaffe

'Sword of the Samurai' heißt ein neues Action-Abenteuer-Rollenspiel von Microprose. Der Spieler wird hierbei in das alte Japan entführt, in dem noch einzelne Länder existieren, die sich gegenseitig um die Vormachtstellung streiten. In der Rolle eines Samurai muß der Spieler für Ruhe und

Ordnung sorgen, und dies unter anderem durch aussagekräftige Kampfszenen.

'Sword of the Samurai' ist bis jetzt zwar nur als PC-Version erschienen, eine Amiga-Version wird jedoch aller Voraussicht nach ebenfalls bald erhältlich sein.



Sword of Samurai: Kämpfen wie ein echter Samurai



**Z**um Umbau eignet sich allerdings nur der A500 durch seine kompakte breitflächige Bauweise, dies sei schon mal vorangestellt.

Wie jedoch geht nun das Ganze vonstatten?

AMIGA DOS sprach mit Gubert-Ronald Unz, dem Chef der G.R.Unz-Antennenbau GmbH:

AMIGA DOS: Herr Unz, wie sind Sie auf die Idee gekommen, ein Umbauset für den Amiga herauszugeben, welches diesen befähigt, Satellitenprogramme auf handelsüblichen Monitoren auszugeben?

G.R.Unz: Eigentlich war es ein Unglücksfall, der mich auf die Idee brachte. Beim Bau meines Wohnhauses legte ich naturgemäß mit Hand an, Sie wissen schon, man hat Elektriker gelernt und kann damit ganz schön Kosten sparen... Nun, beim Einrichten meines Computerraums hatte ich den Amiga aus meiner alten Wohnung mitgebracht und fing abends nach Einbruch der Dunkelheit an, Steckdosen für die einzelnen Geräte zu setzen. Leider konnte ich die Sicherung nicht herausnehmen, da ich sonst im Dunkeln gestanden hätte. Es wäre ja auch nicht weiter tragisch gewesen, wenn meine Frau, die leider etwas tölpelhaft ist, ääh, das sollten Sie jetzt bitte nicht abdrücken (bitte streichen, die Red.), nicht just in dem Augenblick mit einer frischen Gemüsesuppe hereingekommen und über das Netzteil des Amiga gestolpert wäre. Ich versuchte, den Kochtopf zu fassen, bekam jedoch nur den Deckel, leider aber auch das offene Stromkabel zu fassen. Mit der anderen Hand lag ich gerade auf dem Abschirmungsblech meines A500, den ich aus Reparaturgründen geöffnet hatte.

Nun, so geschah's! Der Strom floß über den Kochtopfdeckel durch meine linke Hand, trat am linken Ohr mit einem Lichtblitz aus, der wiederum führte um meinen Hinterkopf herum, drang am rechten Ohr ein, floß als normaler Strom zu meiner rechten Hand und gelangte von da aus in das Abschirmblech. Im gleichen Moment hörte ich den Schrei meiner Frau: "Da, RTL Plus!" Leider sprach im gleichen Augenblick die Sicherung an,

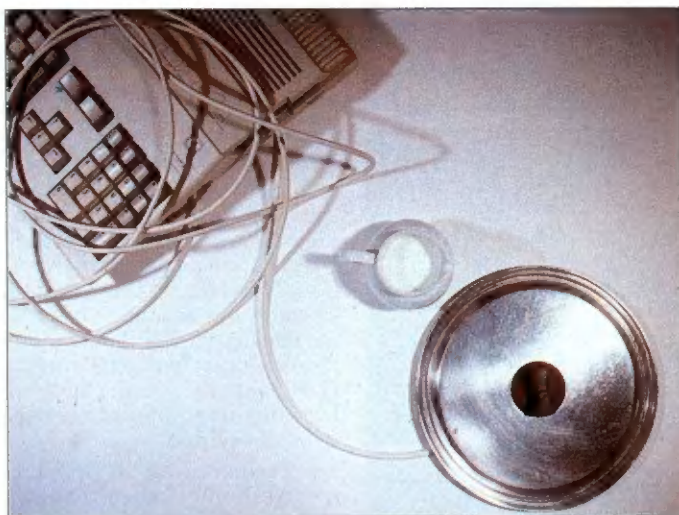


Bild 1. Die benötigten Teile: Ein Kochtopfdeckel und Mäusemilch, das Koaxialkabel paßte leider nicht aufs Bild.

## Schüsseltechnologie

### Amiga als Privat-TV-Empfänger

**Die Redaktion der AMIGA DOS war überrascht – Satelliten-TV auf dem Amiga-Monitor durch den Amiga selbst? Etwas Ungeheuerliches schien wahr geworden zu sein, denn die Nachricht einer bisher unbekannten Firma für Computerzubehör, den Amiga mit wenigen Mitteln in einen Satelliten-Receiver zu verwandeln, erschien uns zu gewagt. Doch sehen Sie selbst.**

so daß es stockdunkel wurde. Trotzdem behauptete meine Frau steif und fest, Kabelfernsehen gesehen zu haben.

Den eigentlichen Versuch wollte ich nicht noch einmal wiederholen, aus verständlichen Gründen wohl gemerkt, aber es mußte einen Weg geben, diesen Vorgang noch einmal zu erzeugen. Und dann fiel es mir wie Schuppen von den Haaren...

ADOS:...von den Fischen...

G.R.Unz:...mir doch egal! Also plötzlich wußte ich, was geschehen war. Der Deckel war die Satelliten-Schüssel, der Lichtkranz um meinen Hinterkopf der Verstärker. Ich begann zu experimentieren und schon bald hatte ich die komplette Schaltung entwickelt!"

Soweit G.R.Unz. Wir, die Redaktion der AMIGA DOS, haben es geschafft, die Erlaubnis für einen Abdruck der

Schaltung zu erhalten. (Bild 1).

### SAT 1 und zufrieden

Was passiert nun eigentlich im Rechner? Zuerst ein Hinweis: Weil man niemandem zumuten kann, durch 220 Volt aus der Steckdose einen Lichtkranz um den Hinterkopf zu erzeugen, ist es notwendig, für entsprechenden Ersatz zu sorgen. Für eine gleichwertige Verstärkung benötigt man auf den Zentimeter genau Koaxialkabel mit einer Länge von 2869,73 Meter. Dieses Kabel wird mit etwas Lötzinn und einem LötKolben an den Kochtopfdeckel gelötet und zwar so: Seele (der Mitteldraht) an den Griff, die Abschirmung an den Rand. Als Kochtopfdeckel eignen sich nur Einmachtopfdeckel mit einem Mindestdurchmesser von 2,23 m. Der Kochtopfdeckel

muß nun in einem Abstand von 10,73 Metern vom Amiga aufgestellt werden, die restlichen 2859 Meter werden um den Amiga in einem Umkreis von 82 cm als Spule gelegt. Daraus ergibt sich eine Spulenhöhe von 7,43 Metern. Durch Einschalten des Amiga wird nun eine sogenannte Umdenkspannung erzeugt, die der Streustrahlung des Prozessors um 180 Grad Celsius entgegengerichtet ist. Da diese Umdenkspannung gefährlich für den Prozessor ist, benötigen wir eine Kanne mit Mäusemilch, die im Abstand von 2 Minuten 26 Sekunden auf das Plastikgehäuse des Prozessors geträufelt wird (Beim 68020 bitte 1 Minute 32 Sekunden einhalten, bei 68030 lohnt sich das Nachgießen nicht mehr). Die Mäusemilch bekommen Sie, liebe Leser ganz einfach: Lesen Sie unseren Bericht, und in Ihrem Kopf entsteht automatisch der Satz: "Das ist doch zum Mäusesmelken!" Tun Sie es doch!!

Durch die Kühlung in Intervallen entwickelt sich eine Hochfrequenz, die sich folgendermaßen berechnet:

$$SP_H = \frac{mm_{Int} * 82 + 2869,73}{PI * D}$$

$SP_H$  = Spulenhöhe (m)

$mm_{Int}$  = MäuseMilchIntervall

$PI$  =  $\pi$

$D$  = Dau(men)

Die Formel ergibt also eine errechnete Frequenz von 31,31 GigaHertz.

Diese Frequenz wird von der Abschirmplatte des A500 reflektiert und auf den Bildschirm des Monitors geworfen und zwar brutal frontal, weshalb sie auch BruFroTal-Frequenz oder  $F_{Latsch}$  (LATente SCHadwelle) genannt wird. Die im Monitor entstehende Bildwiederhol-Frequenz wird mit der Schadwelle gekoppelt und zur Bildisweg-Frequenz, die den vom TV-Satelliten ausgesendeten Signalen täuschend echt gleicht. Der Monitor merkt den Unterschied nicht und gibt ohne Protest jedes Privat-TV-Programm zum besten. Sie können so lange fernsehen, bis Ihnen die Mäusemilch ausgeht oder Ihnen die Seifenopern zum Hals raushängen.

Das AMIGA-DOS-Team wünscht Ihnen viel Glück beim Nachbau.

(A.Pril/jb)



# Festplatten

19' ms

750 KB/S

**2 Jahre Garantie**  
**AutoBoot**  
**sehr leise**

  
**Quantum**

**42 MB 1498.-    105 MB 1998.-**



**Wechselplatte**

SCSI, 44 MB, 25' ms, 500 KB/S

**2098.-**

**Testauszug AMIGA (9/89, Seite 151) :**

BOIL! ist einer der schnellsten Festplattentreiber, die für den AMIGA verfügbar sind.

**Testauszug Kickstart (10/89, Seite 19) :**

Insgesamt machen die Festplatten von FSE einen sehr guten Eindruck, was nicht nur an dem hervorragenden BOIL! Treiber liegt, sondern am ganzen Konzept

Obige Preise gelten für AMIGA 2000. Festplatten für AMIGA 500/1000 sowie andere Kapazitäten (84, 120, 170, 210 MB) auf Anfrage.

Damit Sie objektive Werte und nicht nur Herstellerangaben vergleichen können :

Lesegeschwindigkeit unserer Festplatten nach DiskPerfa (Kick PD 170) : 620 KB/Sek.

1) Herstellerangabe

2) Hardware Datentransferrate auf dem Bus

## TEAC Diskettenlaufwerke

Vollkompatibel, anschlussfertig, abschaltbar, inkl. Kabel, amigafarbenem Metallgehäuse, 2x80 Spuren, alle Laufwerke mit beiger Frontblende. Wir verwenden nur Markenlaufwerke von **TEAC** (FD 235 F oder FD 55 GFR). Alle 5.25" Stationen werden mit **40/80 Umschaltung, durchgeführtem Bus und original Commodore - Treiberplatine** geliefert. Auf alle TEAC Diskettenlaufwerke geben wir **1 Jahr Garantie**. Durchgeführter Bus (bei 5.25" Serie) : DM +10.-

**3.5": 229.-**

**5.25": 259.-**

Unsere Produkte erhalten Sie auch auf folgenden Messen :

**HobbyTronic**  
25. - 29.4.90

**AMIGA**  
9. - 12.5.90

**Dortmund**

**Basel**

**FSE**

**Frank Strauß Elektronik**  
**Schmiedstraße 11**  
**6750 Kaiserslautern**  
Tel.: (0631) 67096 - 98  
Fax: 60697

Donnerstags bis 20.30 h geöffnet.



Ralf Zuber

# Deluxe Video III

## Video mit dem Amiga

**Das Raumschiff steht startbereit auf der Rampe. Unter ohrenbetäubendem Lärm zünden die Triebwerke. Der Rauch der Triebwerke verdichtet sich und nimmt den Zuschauern die Sicht. Langsam erhebt sich das Schiff und fliegt dem blauen Himmel und seinem Ziel entgegen.**

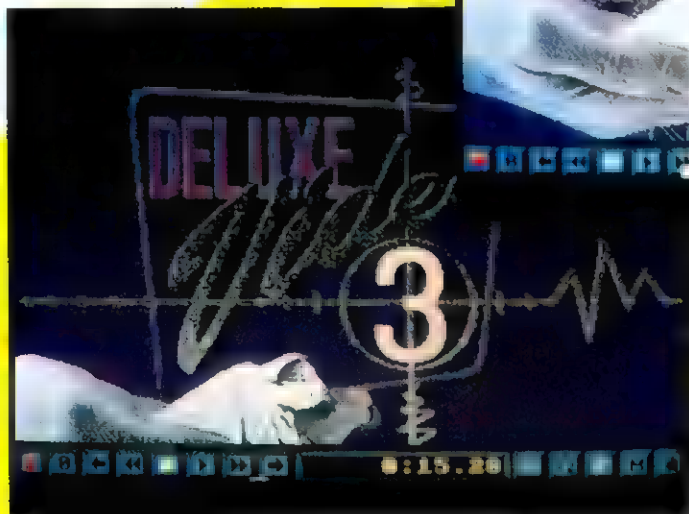
**N**ein, wir sind nicht in Cape Canaveral, und wir sind auch keine Zuschauer eines Videofilms. Aber so wie das eben beschriebene Szenario könnte ein Videoclip aussehen, den Sie mit Deluxe Video III, einem Amiga mit genügend Speicher und etwas Phantasie selbst erstellen können. Ohne bewegte Bilder würde der Amiga nur halb soviel Faszination auf seine Anwender ausüben. Doch von kleinen Bewegungen in Bildern, die mit den meisten Anima-

tionsprogrammen realisierbar sind, bis zu einem fertigen Videoclip ist es ein ganzes Stück Arbeit. Deluxe Video III präsentiert sich als Programmpaket, dessen Leistungen weit über einfache Animation hinausreichen. Vielmehr liegt hier ein Produkt vor, mit dem Videos von A bis Z, also in

Sind Sie Besitzer einer Festplatte, können Sie das Programm auch problemlos dort installieren, da auf einen Kopierschutz verzichtet wurde.

Um einer solch komplexen Aufgabe wie der Erstellung eines Videos gerecht zu werden, bedarf es mehr als nur eines einzelnen Programms. Dementsprechend besteht Deluxe Video aus mehreren Programmteilen. Wer ein Video erstellen möchte, muß sich zunächst einmal über den Inhalt des geplanten Streifens klar sein. Mit DV-Maker (Deluxe Video Maker) kann das Drehbuch im Detail vor-geplant

anderen einblenden, es langsam aufbauen oder die einzelnen Farben aus einem schwarzen Screen heraus-schälen. Über die eben beschriebenen Möglichkeiten gibt es noch eine ganze Reihe weiterer Effekte, um Bilder ein- und auszu-blenden. Auch das Überblenden von einem



ihrer Gesamtheit aus Bewegung, Musik und Szenenabfolge, erstellt werden können.

Jedoch muß Ihr Amiga einige Voraussetzungen erfüllen, um Deluxe Video III benutzen zu können. Es wird ein Minimum von 1 MByte Speicher und ein zweites Diskettenlaufwerk benötigt.

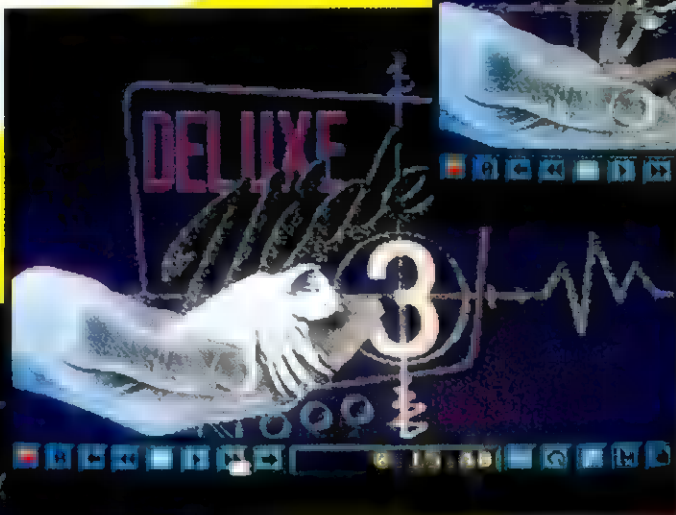


Bild zum anderen stellt kein Problem dar. Die zur Verfügung stehenden Effekte werden über ein Effektrequester angezeigt und können mit einem Mausklick ausgewählt werden.

werden. Aus diesem Videoscript sind alle Szenen und Effekte ersichtlich. Was das Videoscript für das gesamte Video, ist das Szene-Script für die einzelnen Szenen. Hier muß der Anwender dem Computer mitteilen, was im einzelnen passieren soll. Ein einfaches Beispiel: Die erste Szene eines beliebigen Videoclips soll das digitalisierte Bild einer Frau sein. Nun muß im Szene-Script definiert werden, daß das betreffende Bild, bevor es gezeigt wird, erst einmal von einer Diskette eingelesen werden muß. Ist das Bild im Speicher, kann der Amiga es einfach von einem Sekundenbruchteil auf den

Das Programm erlaubt die Benutzung von allen Auflösungen und unterstützt auch den HAM-Modus. Die Größe eines Bildschirms kann bis zu 1024x1024 Punkten betragen. Die Integration von Animationen, die unter Deluxe Paint erstellt wurden, ist problemlos möglich. So richtig Bewegung kommt mit der Benutzung von Brushes auf den Bildschirm, wobei auch hier wieder Animationen und Colorcycling unterstützt werden. Hier offenbart Deluxe Video, wie leicht das Erstellen von Videos sein kann. Wollen Sie zum Beispiel einen Brush von der linken zur rechten Ecke wandern lassen, brauchen Sie nur den Start- und Endpunkt mit der Maus zu be-





stimmen, und den Rest übernimmt das Programm. Doch auch schwierigere Bewegungsabläufe stellen beim DV-Maker kein Problem dar. Nehmen wir mal an, Sie wollen einen im Bogen geworfenen Ball darstellen. Nun müssten Sie normalerweise mehrere Koordinaten der Flugbahn angeben. Nicht so beim DV-Maker. Sie bestimmen wie im ersten Beispiel den Startpunkt und zeichnen dann einfach mit der Maus die Flugbahn, wobei auch die Geschwindigkeit berücksichtigt wird. Das heißt, wenn Sie am Anfang die Maus schneller über den Bildschirm bewegen als am Ende, so wird Ihr Ball im Video das gleiche tun. Doch was ist ein Video ohne Geräusche? Auch hier hilft uns das Programm weiter. Es erlaubt die Verwendung von Sounds, die im 8SVX-Format vorliegen müssen.

## Soundtracks – Action auch akustisch

Ein Film ohne Musik ist wie eine Currywurst ohne Ketchup. Auch die Deluxe-Video-Macher haben eingesehen, daß Stummfilme wenig konkurrenzfähig sind. Dementsprechend enthält Deluxe Video III auch Optionen, um Musik und Geräusche in die Videos einzubinden. Diese

müssen als IFF-SMUS-File vorliegen.

Auch die Darstellung von Text wird unterstützt. Über einen entsprechenden Requester können Sie unter mehreren verschiedenen einen Zeichensatz auswählen und benutzen, und sogar 'Special Effects' wie Schatten können leicht erreicht werden. DV-Maker erlaubt auch die Verwendung von Genlock und Midi-Instrumenten. Damit dürfte man schon qualitativ hochwertige Videos erstellen können. Nun noch eine kurze Erläuterung zu den anderen Programmen. DV-Player ist ein Programm, das Ihre erstellten Videos abspielt. Das Erfreuliche daran ist, daß dieses Programm Public Domain ist. Somit können Sie Ihre Videos auch an Bekannte schicken und von diesen bewundern lassen. DV-Mover kopiert alle für ein Video benötigten Dateien, zum Beispiel Bilder, Brushes oder

Sounds, von den verschiedenen Disketten, so daß sich dann das Video mit allen Dateien zusammen auf einer Diskette befindet. Und zu guter Letzt gibt es noch ein Programm, mit dem man sich eine Dia-Show zusammenstellen kann.

## Videos grenzenlos?

Haben Sie Animationen mit DPaint III erstellt, können Sie diese genauso verwenden wie Raytracing-Bilder mit 4096 Farben. Die Möglichkeiten, mit Deluxe Video III seine eigenen Videos zu erstellen, sind mannigfaltig und nach kurzer Einarbeitungszeit leicht zu bewältigen.

Neben einer logischen und übersichtlichen Benutzerführung besticht das Programm mit einer, wie bei Electronic-

Arts-Produkten üblich, ausgezeichneten Dokumentation. Das Buch enthält neben detaillierten Erklärungen aller Programmteile auch ein Tutorial, in dessen Rahmen der Benutzer anhand klarer Beispiele in die Deluxe-Video-Benutzung eingeführt wird. Programme und Handbuch sind komplett in Englisch und damit das einzige, echte Manko, das der deutsche Benutzer hier in Kauf nehmen muß.

(R. Zuber/hs)

## AMIGA DOS Blitzlicht

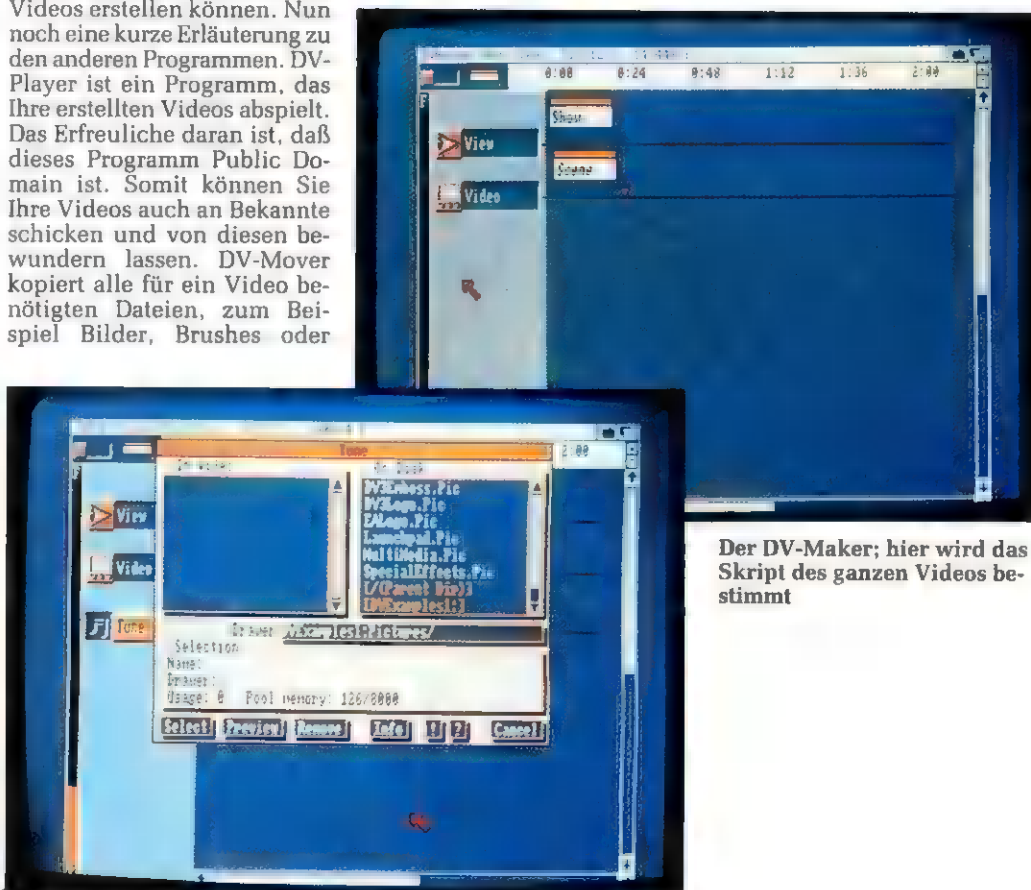
**Name:** Deluxe Video III  
**Hersteller:** Electronic Arts  
**Vertrieb:** Fachhandel  
**Preis:** voraussichtlich zirka 300 DM  
**Besonderheiten:** mindestens 1 MByte Speicher und zweites Laufwerk erforderlich

### Positiv:

- klare Benutzerführung
- leicht erlernbar

### Negativ:

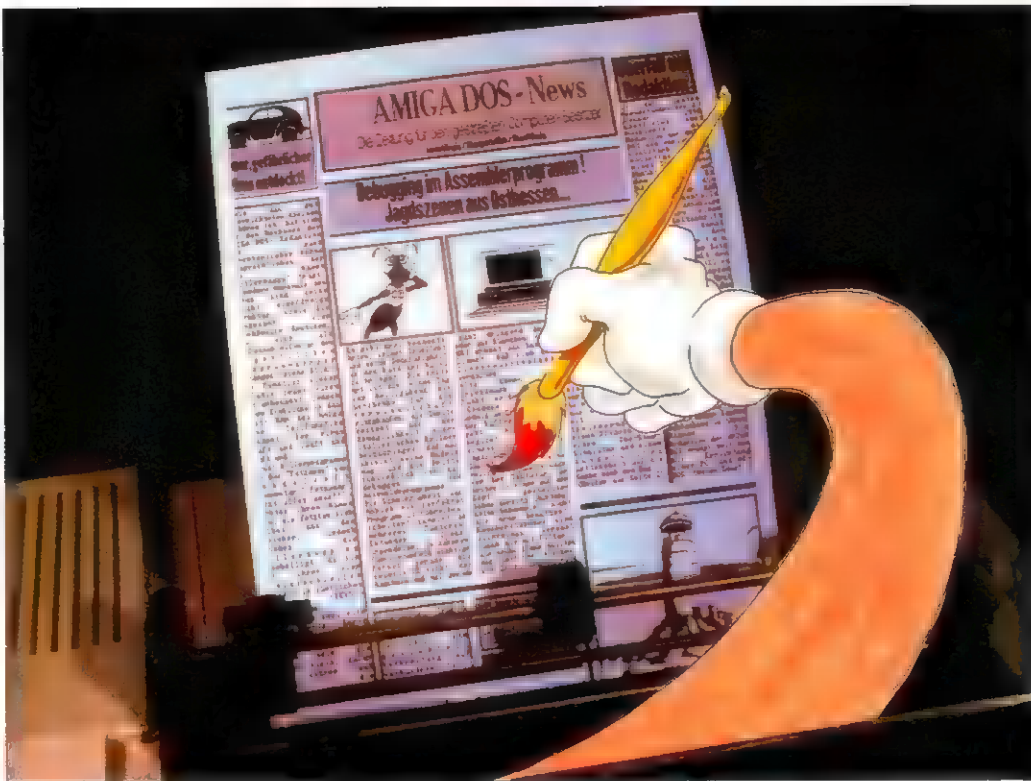
- Handbuch und Programm komplett in Englisch



Der DV-Maker; hier wird das Skript des ganzen Videos bestimmt

Mit Preview können Videos, die in der Entwicklung sind, unter die Lupe genommen werden





Andreas Polk

## Schönschrift für den Matrixdrucker

### Der Postscript-Interpreter für den Amiga

Desktop Publishing ist zwar inzwischen auch für den Heimbereich attraktiv geworden (verschiedene Programme beweisen das), jedoch wird die Ausdruckqualität durch den Umstand gemindert, daß an den meisten 'Heimcomputern' (zu denen inzwischen auch der Amiga und PC gehören) ein Matrixdrucker seine Dienste verrichtet.

**D**ies liegt jedoch weniger an dem Umstand, daß die Programme nicht gut sind oder DTP nicht für den Heimbereich zu gebrauchen ist, sondern daran, daß die Druckdateien für die entsprechenden Seiten im PostScript-Format vorhanden sind. PostScript ist eine spezielle Sprache für High-Tech-DTP-Anlagen wie die hier im Heft vorgestellte. Matrixdrucker können mit PostScript-Dateien soviel anfangen wie der Eskimo mit einem Kühlschrank (jedenfalls solange er noch in den Kälteregeionen der Erde lebt). Was braucht also der DTP-Begeisterte, der einen vernünftigen Ausdruck auf seinem Matrixdrucker haben will? Richtig, einen PostScript-Interpreter. Und damit wären wir beim Thema.

Die einzigen Drucker, die die Sprache PostScript im Au-

genblick verstehen, sind die entsprechenden Laserdrucker. Mit PixelScript soll sich dies nun ändern, denn es können alle Postscript-Dateien auf den gängigen Preferences-Druckern ausgedruckt werden.

### Wie sag ich's meinem Drucker?

Zuerst einmal zu der Frage, was PostScript überhaupt ist. PostScript ist eine Sprache, die nur für die Druckeransteuerung konzipiert wurde. Der Drucker wird nicht mehr über Kontrollsequenzen wie beispielsweise den ESC/P-Standard angesprochen, sondern über eine eigene Programmiersprache. Der Unterschied zu anderen Programmiersprachen besteht darin, daß PostScript im Gegensatz zu C oder einer anderen Spra-

che nur für die Ausgabe von Daten auf den Drucker zuständig ist. Somit besitzt diese Sprache sehr viele nützliche Befehle, und der Ausdruck ist entsprechend gut. Der Nachteil an der ganzen Angelegenheit ist jedoch der, daß nur sehr teure Drucker – in der Regel Laserdrucker ab 10000 DM – diese Sprache verstehen, denn dazu ist ein PostScript-Interpreter notwendig. Dies ist eine Hardware, die im Drucker eingebaut ist und nur die Aufgabe hat, die PostScript-Datei zu interpretieren und auszudrucken. Diese Hardware ist recht teuer, dafür aber auch entsprechend schnell. Es können mehrere Seiten pro Minute interpretiert und gedruckt werden. Da aber kaum jemand über einen solch teuren Drucker verfügt (wer hat schon 10000 bis 30000 Mark Kleingeld in der Brieftasche), muß hier Abhil-

fe geschaffen werden. Die preiswerteste Methode besteht darin, den Hardware-Interpreter durch Software zu ersetzen. Somit können nicht nur Laserdrucker, sondern jeder beliebige Preferences-Drucker angesprochen werden.

### PixelScript soll die Lösung heißen

PixelScript ist das erste Programm für den Amiga, das diese Aufgabe bewältigt. Es wandelt also PostScript-Dateien so um, daß sie auf jedem Drucker ausgedruckt werden können, für den ein Preferences-Treiber existiert. Natürlich müssen hierbei Abstriche gemacht werden. Zum einen dauert das Interpretieren einer Seite – also das Umwandeln der PostScript-Datei in eine für den Drucker verständliche Form – viel länger als bei PostScript-fähigen Druckern. Wartezeiten von einer Stunde sind bei komplexen Seiten nicht selten. Außerdem kann der Ausdruck eines 24-Nadel-Druckers auch in der höchsten Auflösung nicht mit dem Ausdruck eines Laserdruckers mithalten.

So, nun aber genug der Einführung. Widmen wir uns dem Programm PixelScript einschließlich seiner Vor- und Nachteile. Zuerst muß PixelScript installiert werden. Hierzu wird ein Installationsprogramm mitgeliefert, das die Installation ausreichend vornimmt. Der einzige Nachteil ist der, daß das Programm DPS, auf das gleich noch eingegangen wird, nicht automatisch mit installiert werden kann. Hier muß man über das CLI gehen. Startet man PixelScript von der Workbench aus, so wird ein neuer Screen geöffnet. In der linken oberen Ecke befindet sich ein Rad, welches anzeigt, ob PixelScript gerade rechnet oder nicht. Nach kurzer Zeit kann der Benutzer Befehle eingeben.

Bevor man nun erwartungsvoll den ersten Ausdruck startet, sollte man erst einmal schauen, ob PixelScript auch richtig konfiguriert, also an die vorhandene Hardware angepasst ist. Hierzu wählt man den Menüpunkt 'Configure' aus. Es erscheint ein neues Fenster mit vielen Einstellungsmöglichkeiten. Neben den Einstellungen, die schon durch Preferences festgelegt sind, sind noch einige zusätzliche möglich.



## PixelScript

Abb. 1. Probeausdruck  
360\*360 dpi mit Excellence!...

So kann beispielsweise festgelegt werden, daß der Druckbereich des Blattes kleiner ist als in Preferences angegeben. Auch ein optionaler Treiber kann bestimmt werden. Besitzen Sie nämlich einen nicht PostScript-fähigen Laserdrucker, so brauchen Sie nicht den Umweg über den (nicht vorhandenen) Preferences-Treiber zu gehen, sondern können hier einen Treiber auswählen.

### Die Konfiguration des PostScript-Umwandlers: Aller Anfang ist schwer

Für diesen Treiber können dann noch andere Angaben gemacht werden, wie beispielsweise die Festlegung der Druckauflösung. Möchten Sie IFF-Dateien drucken, so können Sie in dem Gadget Screen-Frequency noch zusätzliche Einstellungen vornehmen. Auch der Rasterwinkel einer Grafik, die sich aus Graustufen zusammensetzt, ist einstellbar.

Als Besonderheit bietet PixelScript ferner die Möglichkeit, eine Datei nicht auf den Drucker auszudrucken, sondern in eine Datei zu schreiben. Dies ist vor allem dann sinnvoll, wenn Sie sich eine PostScript-Datei erst einmal anschauen wollen, bevor Sie den Druck starten. Um eine so entstandene Datei anzuschauen, befindet sich das Public-

## PixelScript

Abb. 2. ...und mit PixelScript (Originalgröße)

Domain-Programm DPS im Lieferumfang. So lassen sich mögliche Fehler leichter finden, da nicht erst ein Ausdruck, der wesentlich länger dauert als das 'Drucken' in eine Datei, gestartet werden muß. Außerdem können Sie von den entstandenen Grafiken Hardcopies anfertigen und in Präsentationsprogrammen weiterverarbeiten.

Möchten Sie eine bereits in eine Datei geschriebene Datei nachträglich ausdrucken lassen, so brauchen Sie die PostScript-Datei nicht noch einmal durch den Interpreter zu schicken. Das Programm PxSPrint übernimmt diese Aufgabe. Allerdings druckt es nicht in der höchsten Auflösung, sondern in der, die beim Druck in eine Datei angegeben wurde. Da dies meistens 72\*72 dpi ist, wird man in den meisten Fällen um ein nochmaliges Interpretieren nicht herumkommen, möchte man das beste Ergebnis erzielen. (Das Programm PxSPrint darf übrigens frei kopiert werden.)

Haben Sie alle Einstellungen vorgenommen, so können Sie diese unter einem speziellen Namen abspeichern. Auch das automatische Laden Ihrer Konfigurationsdatei beim Start von PixelScript ist möglich. Haben Sie die Konfiguration eingestellt, so müssen Sie nur noch die zu druckende PostScript-Datei auswählen. Hierzu gibt es eine File-Select-Box. Nach Auswählen einer Datei beginnt Pixel-

Script mit dem Interpretieren der Datei und druckt sie dann aus. Dieser Vorgang kann bei ganz einfachen Seiten recht schnell gehen, bei komplexen Seiten können aber Wartezeiten von zirka einer Stunde auftreten.

### Ein starkes Gespann gegen PostScript-unwillige Drucker

Dafür läßt sich das Ergebnis sehen. Mit einem 24-Nadel-Drucker in der höchsten Auflösung von 360\*360 dpi gleicht das Ergebnis schon stark dem Ausdruck eines Laserdruckers. Jegliche Rundungen der Buchstaben erscheinen auch tatsächlich rund. Der Treppchen-Effekt entfällt fast völlig. Nur bei ge-

nauem Hinschauen sind einzelne Punkte, die allerdings durch die Größe der Nadeln bedingt sind, erkennbar. Da PixelScript die Druckertreiber der Workbench benutzt (es sei denn, Sie besitzen einen Laserdrucker und nehmen die PixelScript-eigenen Treiber), ist es möglich, das Programm TurboPrintII, welches die Workbench-Treiber durch eigene, bessere ersetzt, zu nutzen. Somit kann der Ausdruck noch einmal beschleunigt werden. Hier ist das Programm TurboPrintII eine zu empfehlende Ergänzung zu PixelScript.

Sollte einmal ein Fehler aufgetreten sein, so erscheint ein Requester, der die Fehlerart anzeigt. Der geübte Anwender kann hier nun Korrekturen vornehmen und den Aus-

cken Sie PostScript-Dateien auf Ihrem  
ferences-Drucker.

# PixelScript

Versio  
Der PostScript-Interpreter für den Am

Schriftpro

Helvetica 10p  
Helvetica 12  
Helvetica 1  
Helvetica 1

#### Textprobe

**Einführung**  
PixelScript ist der PostScript-Interpreter für den Amiga Computer. Er ermöglicht es Ihnen, PostScript-Dateien auf Ihrem Matrix-, Tintenstrahl- oder Laserdrucker in der höchsten Auflösung anzugeben. Solche PostScript-Dateien werden von vielen DTP- und Textprogrammen erzeugt.  
PixelScript können Sie eine Druckqualität erreichen, die sonst nur auf PostScript-Druckern möglich ist. PixelScript ist der leistungsfähigste Interpreter, der die meisten Fähigkeiten von PostScript

**Gold Vision**  
Communications  
Kurfürstendamm 144 100  
1000 Berlin 15  
Tel.: (030) 88 33 505

Seite wurde mit PageStream gesetzt, als PostScript-Datei ausgeg  
mit PixelScript gedruckt.

#### Einige Tips zum Arbeiten mit PixelScript

An dieser Stelle sollen Ihnen vier kleine Tips gegeben werden, die vielleicht helfen, wenn PixelScript mal nicht so arbeitet, wie Sie es möchten: - Sollten unerklärliche Fehlermeldungen auftreten, installieren Sie PixelScript einfach noch einmal. Bei uns half diese Prozedur.

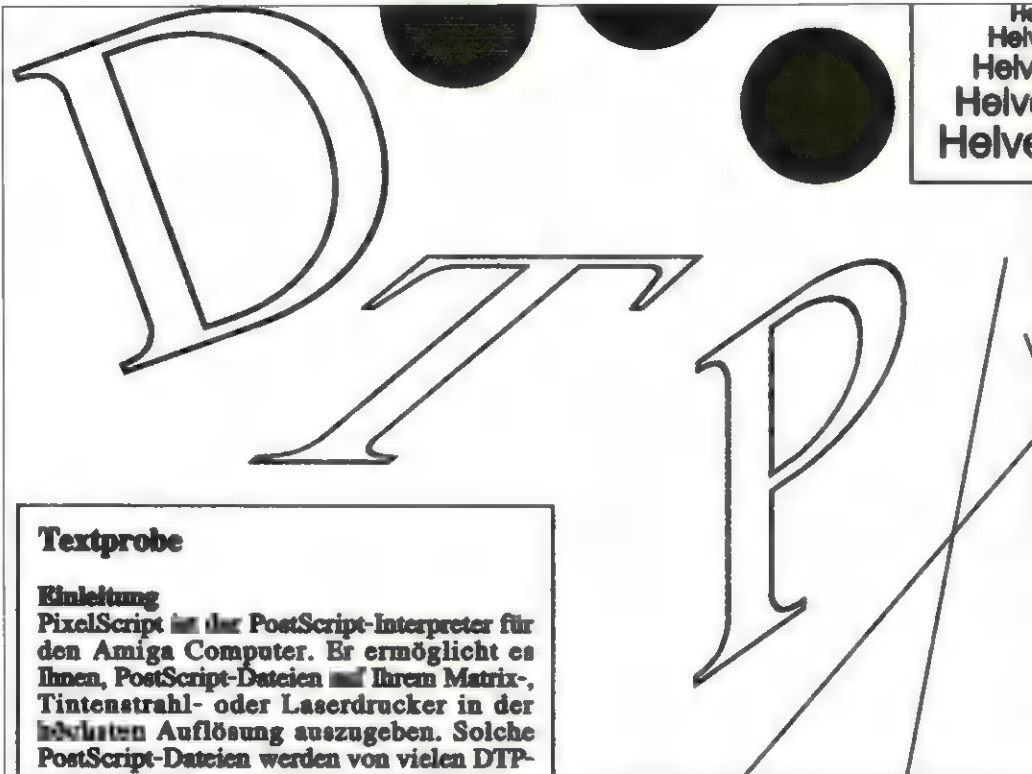
- Bedenken Sie, daß PixelScript die Einstellungen von Preferences benutzt. Ein guter Wert für die Papiergröße ist 8,2 für width und 11,5 für height mit aktiviertem Bounded-Gadget.

- Bedienen Sie PixelScript über das CLI, so müssen Sie die Befehle angeben, wie PixelScript sie im Command-Gadget der Intuition-Oberfläche schreibt. Um einen Ausdruck zu starten, geben Sie beispielsweise (PXS:PXSFiles/Test.ps) run ein. Dies ist auch der Befehl, den PixelScript im Command-Gadget ausgibt, wenn Sie mittels Select-File die Datei PXS:PXSFiles/Test.ps angegeben haben.

- Erscheint immer ein Error, so geben Sie einfach 'Abort' oder 'Resume' an, das normale PixelScript erscheint dann wieder.

Abb. 3. Die Demo-Seite wurde mit PixelScript auf einem 24-Nadel-Drucker ausgedruckt: hier die Verkleinerung...





### Textprobe

#### Einleitung

**PixelScript** ist der PostScript-Interpreter für den Amiga Computer. Er ermöglicht es Ihnen, PostScript-Dateien auf Ihrem Matrix-, Tintenstrahl- oder Laserdrucker in der höchsten Auflösung auszugeben. Solche PostScript-Dateien werden von vielen DTP-

Abb. 4. ...und hier ein Ausschnitt in Originalgröße

druck wiederholen oder einfach den Fehler übergehen. Wenn Sie die Sprache PostScript lernen möchten, ist PixelScript auch hier eine nützliche Hilfe. In einem Command-Gadget können einzelne PostScript-Befehle eingegeben werden. Diese werden dann direkt interpretiert und ausgedruckt. So fällt das Lernen der Sprache wesentlich einfacher, da Befehle ausprobiert und die Ergebnisse direkt angeschaut werden können.

Nun fragt man sich natürlich, wo solche PostScript-Dateien überhaupt herkommen. Man kann sie natürlich selber programmieren. Dies ist jedoch sehr kompliziert und umständlich. Viele Programme bieten die Möglichkeit, PostScript-Dateien zu erzeugen. Dazu gehören die meisten DTP-Programme, wie beispielsweise PageStream oder Professional Page. Auch einige Textverarbeitungen bieten diese Möglichkeit, wie beispielsweise 'Excellence!'.

Beim Test benutzte ich PostScript-Dateien von PageStream und von Excellence!. Hierbei gab es keine Probleme. PixelScript hat diese Dateien alle akzeptiert und gedruckt. Die Qualität des Ausdrucks mit PixelScript ist wesentlich höher als die der Ausdrucke der Programme.

Natürlich hat auch PixelScript einige Nachteile, die hier nicht unterschlagen werden sollen. Zuerst wäre hier der überaus hohe Speicherbedarf anzuführen. Mit 512 kByte kommt man nicht aus, 1 MByte ist das Minimum, reicht aber auch nicht immer aus. Eine 2-MByte-Erweiterung ist zu empfehlen. Möchten Sie Speicherplatz sparen, so bietet PixelScript die Möglichkeit, die Intuition-Oberfläche zu unterbinden. So werden laut Handbuch 170 kByte gespart. PixelScript wird dann interaktiv über das CLI gesteuert.

Hier wären wir auch schon beim nächsten Minuspunkt: das Handbuch. Es erwähnt zwar, daß man PixelScript interaktiv über das CLI bedienen kann, aber wie wird nicht verraten. Der Benutzer muß die Befehle selbst herausfinden. Von einem Programm, welches 300 DM kostet, kann man schon ein komplettes und ausführliches Handbuch erwarten. Zuletzt ist eigentlich nur noch der Preis zu nennen. 298,- DM für ein Utility, denn das ist PixelScript ja letztendlich, ist recht viel. Hier muß der Anwender selber entscheiden, ob er diese Summe investieren möchte. Zuerst muß man sich im klaren sein, daß PixelScript seine hauptsächliche Anwen-

dung im Desktop-Publishing-Bereich findet. Damit ist ein DTP-Programm (oder eine Textverarbeitung, die in der Lage ist, PostScript-Dateien zu erzeugen) schon fast ein Muß. Ist man dabei jetzt auf guten Ausdruck angewiesen, wird man diese Summe sicherlich zahlen.

### Die Testergebnisse zusammengefaßt

Summiert man alle Vor- und Nachteile, so kommt man zu dem Schluß, daß PixelScript ein gutes Programm ist, wel-

ches seinen Zweck sehr gut erfüllt.

Die Druckergebnisse sind auch mit einem Nadeldrucker sehr gut. Die Geschwindigkeit läßt sich durch gleichzeitiges Verwenden von TurboPrintII noch beschleunigen. PixelScript zielt allerdings mehr auf den gehobenen Anwendungsbereich, da es sehr viel Speicher benötigt und auch nicht gerade billig ist. Der vorhandene ARexx-Port läßt hoffen, daß jemand einmal eine Schnittstelle zu einem Anwenderprogramm definiert. Somit könnten Ausdrucke direkt von Anwenderprogrammen aus gestartet werden!

(jb)

### AMIGA DOS Blitzlicht

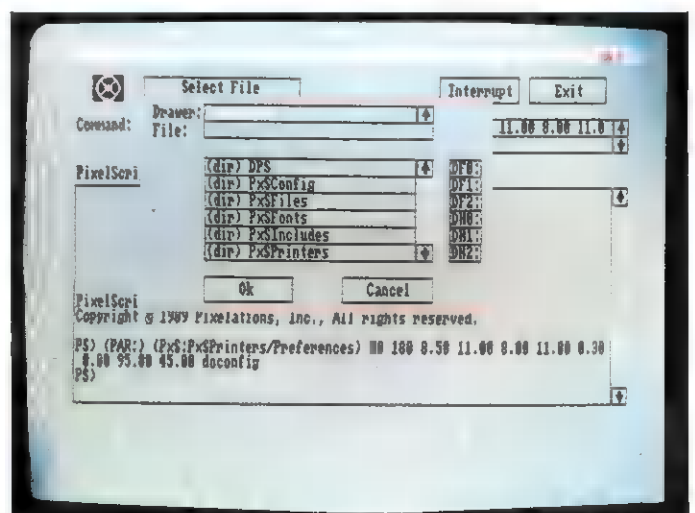
**Name:** PixelScript  
**Anbieter:** Gold Vision, Kurfürstendamm 64-65, 1000 Berlin 15, Tel.: 030/8833505  
**System:** Alle Amiga ab 1 MByte (je mehr Speicher, desto besser)  
**Besonderheiten:** Drucken in Datei, eigene Druckertreiber für Laserdrucker, zusätzliche Utilities

#### Positiv:

- sehr guter Ausdruck
- Eingabe einzelner PostScript-Befehle möglich
- Konfiguration abgespeicherbar
- Starten ohne Intuition möglich (Speicher sparen)
- Unterstützung von ARexx

#### Negativ:

- schlechtes Handbuch
- viel Speicher nötig
- hoher Preis



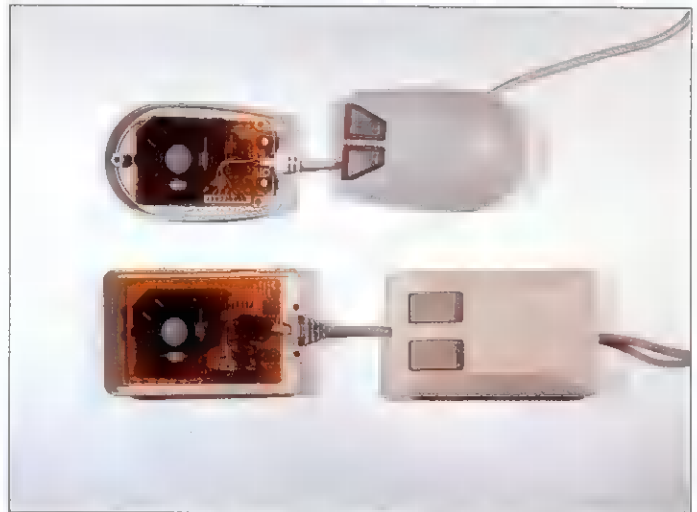
Der 'Configure-Screen' von Pixelscript



# Mäusemächtige Eingabemedien

## Reisware- und Amiga-Maus im Wettstreit

Wer den Amiga gerne unter Intuition benutzt, kommt ohne Maus nicht aus. Bisher waren Amiga-Anwender bei einem Versagen des rollenden Eingabemediums auf original Commodore-Ersatzteile angewiesen. Mit der Reisware-Maus gibt es nun eine echte Alternative.



Beiden Mäusen in den Bauch geschaut. Das Innenleben beider Geräte unterscheidet sich nur unwesentlich voneinander

**O**val, in kargem Grau gehalten, macht die Reisware-Maus auf den ersten Blick einen schlichten Eindruck. Jedoch braucht sich die Reisware-Maus auch nicht hinter Äußerlichkeiten verstecken. Die ovale Form ist sehr grifffreundlich und der Anatomie der Handinnenflächen angepaßt. Die Maustasten sind,

wie auch bei der original Commodore-Maus, an der Vorderseite angebracht.

Was das Innenleben angeht, unterscheiden sich die beiden Mäuse nur unwesentlich voneinander. Beide weisen die obligatorischen Mikroschalter als Maustasten auf (ältere Amiga-Mäuse sind teilweise noch nicht mit Mikroschaltern ausgerüstet). Für

das restliche Innenleben zeichnet sich bei beiden Mäusen Mitsumi Elec. verantwortlich.

Tatsächlich unterscheiden sich beide Mäuse im wesentlichen durch das Design, das bei der Reisware-Maus anwenderfreundlicher gestaltet wurde.

Die Reisware-Maus kostet 89,- DM und genießt einen zwölfmonatigen Garantieschutz.

Die Maus von Commodore kostet 99,- DM und hat sechs Monate Garantie.

(hs)

# ABC-SOFT

NEUE ANSCHRIFT  
Hangstein 16a · D-4920 Lemgo  
Tel. 05261/68475 Fax 05261/68229

Fachversand für AMIGA Hard- und Software — Public Domain — Shareware

**Über 4.500 PD-Disks aus ca. 120 Serien**  
**■ Zeit lieferbar!**

z.B. Fred Fish, Kickstart, Taifun, ACS, RPD, Chiron, Faug, RHS, Auge, Tornado, Pornoschow, Cactus, TBAG, Panorama, Safe ....

**2.10 DM** je 3,5" PD-DISK  
bei Abnahme von 100 Stück.

**2.20 DM** bei Abnahme von 1 - 99 Stück  
Preise inkl. 3,5"-Disk / -inkl. Etiketten / - mit doppeltem  
»Verify« auf 1a NoName-Disks kopiert.  
auf 5,25" 1.40 ■ bei 1 - 49 St. 1.20 ■ ab 50 St.

■ BITTE KOSTENLOSES INFO ANFORDERN! ■

**ABO-MÖGLICHKEIT** auf Neuerscheinungen aller oder nur bestimmter Serien!

**3 DEUTSCHE KATALOGDISKS 10.- DM**  
(Scheck, Briefmarken oder bar) zzgl. 2.50 DM Porto.

**SPIELE-PAKET 1 40.- DM**  
ca. 30 PD-Spiele auf 10 Disks

**SPIELE-PAKET 2 49.- DM**  
11 PD-Spiele der Spitzenklasse auf 10 Disks

**EINSTEIGER-PAKET 40.- DM**  
für Amiga-Anfänger mit CLI-Hilfen, Infos, Demos usw.  
auf 10 Disks

**SUPER-PAKET 55.- DM**

bestehend aus Textverarbeitung, CAD, Haushaltsprogramm, Anti-Virus-Disk mit 15 Viruskillern, Spiele und nützlichen Utilities, (Test in der Zeitschrift PUBLIC-DOMAIN 5/89, Urteil: Die Qualität ■ Programme ist gut ■ sehr gut) - ■ Disks

**NEU! DELUXE-BENCH 29.90 DM**

Eine Superdisk zum Einführungspreis!!! Endlich ist komfortables Arbeiten mit dem Amiga und CLI möglich! 1,3 MB der besten AMIGA-Arbeitshilfen in komprimiertem Format. Bereits beim Booten wird die neueste Version von VirusX, die resistenteste Ramdisk (VDO) und ein Anti-Guru-Programm im System installiert. Weitere Utilities: 3 schnelle Kopierprogramme für ■ zu 4 Laufwerke, Boot-Intro-Maker, Mausbeschleuniger, Textverarbeitung, Bildschirmschoner, ein- und ausschalten des Audio-Filters, Packer/Entpacker mit Maussteuerung, Utili-Master z. Ausführen aller CLI-Befehle ■ Mausclick usw. ....

## Oktalyzer

**DM 99.-**

Dieses Programm setzt im Bereich Musik neue Maßstäbe. Es ist MIDI-fähig und besitzt eine Option ■■ Sampeln, wobei die Samples ■ allen möglichen Variationen manipuliert werden können. Sensationell ist die Fähigkeit, **echte ■ Stimmen** gleichzeitig wiedergeben zu können.

## PC Handler

**DM 69.-**

konvertiert MS-DOS- und Atari-Dateien ins Amiga-Format und umgekehrt. Dies betrifft sämtliche DOS-Kommandos. Geeignet für 5.25" und 3.5" Disketten. Keine PC-Karte und kein PC-Laufwerk erforderlich! Konvertiert auch Zeichensätze und IFF-Grafiken!

## Glücksrad

**DM 49.-**

Perfekte Umsetzung des bekannten Fernsehspiels! Mehrere hundert Begriffe sind bereits integriert und können kinderleicht erweitert werden. Mit Konten, tollem Sound und **deutscher Sprachausgabe!** Ein toller Spielspaß für bis zu vier Personen!

## IFF-Sample-Paket DM 79.-

Über 1.000 Samples (Instrumente) in phantastischer Qualität! Verwendbar für alle gängigen Soundprogramme (z.B. Soundtracker, Oktalyzer, Med). Gratis dazu ein PD-Soundprogramm! **Insgesamt 9 Disketten.**

**LEERDISKETTEN von Sentinel**  
**3,5" NN MF 2DD 135 TPI inkl. Aufkleb.**  
ab 10 St. je **DM 1.30** ab 50 St. je **DM 1.27**  
ab 100 St. je **DM 1.25** ab 500 St. je **DM 1.23**  
andere Formate auf Anfrage!

## FARBÄNDER

Star LC 10 **DM 9.90** Star LC 24/10 **DM 14.50**  
NEC P6/P7 plus **DM 14.95**  
Epson LQ 550/800/850 **DM 11.95**

3,5" Laufwerk intern m. Einbausatz **DM 149.-**  
3,5" Laufwerk extern, durchgef. Bus,  
abschaltb., Amiga-farb. Gehäuse **DM 189.-**  
dto. - jedoch NEC 1037a **DM 209.-**  
512 KB-Erweiterg. m. Uhr, abschb. **DM 179.-**  
5,25" Laufwerk extern  
abschaltbar - 40/80 Track **DM 269.-**  
8 MB-Karte für A 2000, 2 MB best. **DM 899.-**

Autoboot-Filecards für Amiga 2000  
bis 500 KB/Sek.  
20 MB **DM 898.-** 30 MB **DM 998.-**  
47 MB **DM 1398.-** 105 MB **DM 2798.-**  
Kickstart-Umschaltplatte  
für 3 Betriebssysteme **DM 55.-**  
U.-Platine inkl. Kickstart V 1.3 **DM 98.-**  
Kickstart-ROM V 1.3 **DM 69.-**

Unsere Versandkosten: NN 3.- DM / Vorkasse 5.- DM / ab 5 kg nach Gewicht / Ausland nur Vorkasse + 15.- DM / Skandinav. 30.- DM





## Desktop Publishing professionell

### Das Desktop-Publishing-Paket in der AMIGA-DOS-Redaktion

**Ein 'neuer' Amiga in der AMIGA-DOS-Redaktion? Des Rätsels Lösung: Hier wird ein Amiga zum professionellen Arbeitsgerät. Die mitgelieferte Peripherie bringt dies noch besser zur Geltung – Postscript-Laser-Drucker, eine 68020-Karte und 3 MByte RAM – das alles gehört zum neuen Betätigungsfeld des Amiga-Desktop-Publishing.**

**W**er die DTP-Anlage geliefert bekommt, steht zunächst vor einem Platz-Problem: Der Laserdrucker von NEC, ein 'Silentwriter LC890' braucht davon eine ganze Menge. Zusammen mit Amiga und Monitor wird der Platz recht schnell knapp, wenn man nicht von vornherein den Platzbedarf einkalkuliert.

Für wen soll nun diese Anlage sein? Heimanwender werden wohl kaum zu den Abnehmern gehören, es sei denn, man gehört zur schweigenden Mehrheit derjenigen, die mal eben locker ca. 30000 DM mit sich herumtragen.

Für die Heimarbeit ist die DTP-Anlage wirklich nicht

geschaffen. Anders dagegen im Business: Werbeleute, Agenturen, die Präsentationen erstellen, dies sind die potentiellen Käufer einer solchen DTP-Anlage. Und für deren Geldbeutel gibt die Anlage schon einiges her.

#### Ein Rennpferd mit Starallüren – der 68020

Wichtigste Verbesserung zu den 'herkömmlichen' Amigas ist der Einbau eines 68020-Prozessors anstelle des sonst üblichen 68000 in einen Amiga 2500. Die Vorteile des Prozessors sind schnell aufgezählt: Volle 32 Bit Rechenlei-

stung liegen hier vor. Bemerkbar macht sich dies vor allem beim Berechnen der PostScript-Dateien. Die mitgelieferte 68020-Karte enthält außerdem einen 68881-Coprozessor, der den Hauptprozessor bei arithmetischen Operationen noch weiter entlastet.

Der Amiga 2500 enthält ebenfalls schon den neuen Big-Agnus-Chip. 1 MByte direkt adressierbares Chip-RAM ist also direkt vorgesehen. Beim Monitor wird auch nicht gespart; ein hochauflösender Multisync-Monitor übernimmt die Rolle, die bisher dem sonst üblichen 1084 vorbehalten war. Aus gutem Grund, wie wir nachher noch

sehen werden. Schließlich der Laserdrucker; er ist wohl das Aushängeschild der DTP-Anlage. Hier sind vollständig PostScript-fähige Ausdrücke möglich, die auf 'normalen' Matrix-Druckern noch immer Zukunftsmusik sind und auch bleiben werden. Dabei bietet der 'Silentwriter LC890' noch mehr Komfort. Sollte er einmal nicht als PostScript-Drucker benötigt werden, so kann er auf eine Hewlett-Packard-Laserjet-Emulation umgeschaltet werden. Diese Emulation wird sogar von den Preferences-Druckertreibern unterstützt. Somit sind ganz einfache Ausdrücke unter der Shell oder den entsprechenden Anwendungsprogrammen möglich. Die Voreinstellungen des Druckers werden an diesem direkt vorgenommen, ein Menü führt dabei durch sämtliche Funktionen. Für unseren Test beließen wir es bei der parallelen Schnittstelle als Datenempfänger und der PostScript-Bearbeitung.

#### PostScript – Mein Drucker versteht mich besser

PostScript hört sich im ersten Augenblick sehr exotisch an, ist jedoch recht einfach zu erklären: Um Drucker direkt anzusteuern, gibt es die Möglichkeit, die Steuercodes per ESC-Code oder per PostScript zu übermitteln. PostScript ist eine spezielle Computersprache, die nur für Drucker interessant ist. Denn alles was in PostScript geschrieben wird, dient nur der Druckerausgabe. PostScript-Dateien müssen allerdings in ein spezielles Format umgewandelt werden, welches der Drucker als seine Steuerzeichen versteht. Diese Umwandlung geschieht über spezielle PostScript-Interpreter, wobei eine nahe Verwandtschaft zu anderen, bekannteren Computersprachen wieder durch das Wort Interpreter gegeben ist. Diese Interpreter wandeln das von uns geschriebene Druckprogramm in für den Drucker verständliche Werte um.

Warum aber PostScript? Nun, der erste positive Punkt ist der, daß man sich nicht mehr mit den Drucker-Codes herumtun muß, gleichzeitig ist damit eine Anpassung an andere Drucker wesentlich einfacher. Auch unsere Zeitschrift entsteht zum Beispiel



- Entwicklung
- Produktion
- Hardware
- Software
- Service

# HK-Computer

## Ihr Amiga-Spezialist

### AMIGA-Bremse "der Highscore-Killer" DM 39,50

- regelt die Geschwindigkeit stufenlos bis zum Stillstand
- ideal für schnelle Games und Bildschirmfotografie

Neu Neu Neu Neu Neu Neu Neu Neu Neu Neu Neu Neu  
Amiga-Bremse für A500 extern mit LEDs DM 69,-  
Neu Neu Neu Neu Neu Neu Neu Neu Neu Neu Neu Neu

I Achtung! Preisänderung bei Festplatten I Achtung!

### HK-COMPUTER Festplatten

Kapazität	Speed	Filecard	A2000	A500
20MB/31/2"	35ms	898,-	848,-	1048,-
30MB/51/4"	65ms	-	898,-	1098,-
30MB/31/2"	35ms	1048,-	998,-	1198,-
40MB/51/4"	28ms	-	1198,-	1398,-
50MB/31/2"	35ms	1298,-	1248,-	1448,-
60MB/51/4"	28ms	-	1298,-	1498,-

Alle unsere Festplatten werden wahlweise mit ALF V1.6 oder mit Autoboot-Modul ausgeliefert.

AUTOBOOTMODUL für A2000 DM 119,-  
AUTOBOOTMODUL für A500 DM 149,-  
zum Nachrüsten, inkl. Software und Anleitung  
(bitte Harddisk- und Controllertyp angeben)

Festplatten-Controller  
OMTI 5520B für MFM-Platten (20/40MB) DM 149,-  
OMTI 5528B für RLL-Platten (30/50/60MB) DM 189,-

Festplatten-Interface DM 99,-  
Die Adapterplatine paßt den PC-BUS eines Festplatten-  
Controllers an den AMIGA-BUS an.  
(Bitte Rechnertyp angeben)

ALF-Festplattentreiber Version 1.6 DM 98,-  
ALF-Festplattentreiber Version 2.0 DM 198,-

Autoboot-Set MFM  
OMTI 5520B, Autoboot-Modul, Festplatten-Interface, Kabelsatz  
für Amiga 2000 (interne Slotkarte) DM 349,-  
für Amiga 500 (extern mit Gehäuse) DM 379,-

Autoboot-Set RLL  
OMTI 5528B, Autoboot-Modul, Festplatten-Interface, Kabelsatz  
für Amiga 2000 (interne Slotkarte) DM 389,-  
für Amiga 500 (extern mit Gehäuse) DM 419,-

Festplattengehäuse A500/1000 DM 379,-  
Amigafarbenes Metallgehäuse, komplett mit Schaltenteil,  
Lüfter, Anschlußadapter mit durchgeführtem Bus, LEDs

Autoboot-Filecard DM 369,-  
Harddisk-Trägerplatine mit integriertem Autoboot-Modul,  
Controlleradapter, Autoboot-Software

### Disketten

31/2" NoName 2DD, 10St. DM 14,95  
31/2" Verbatim Verex 2DD, 10St. DM 25,-  
31/2" Sentinel 2DD ColorDisk, 10St. DM 29,50  
lieferbar in weiß, gelb, orange, rot und grün  
51/4" NoName 2SD, 10St. DM 5,90  
51/4" Verbatim Verex, 10St. DM 14,90  
Staffelpreise auf Anfrage

### Kick-ROM

- Kickstartumschaltplatine für zwei OriginalROMs DM 49,-  
Kick-ROM mit ROM 1.3 DM 98,-  
OriginalROM 1.3 DM 65,-

### Kickstartumschaltplatine 3fach

- für zwei OriginalROMs und eine Epromversion DM 59,-  
Umschaltplatine mit einem Epromsatz DM 155,-  
Umschaltplatine mit OriginalROM1.3 DM 108,-

### Maus & Joystick-Adapter

DM 44,50  
- für gleichzeitigen Anschluß von Maus und Joystick  
- mit LED-Anzeige  
- alle Maussteuerleitungen elektronisch geschaltet

### "PowerFire" Das Superding!

DM 29,50  
- Dauerfeuerinterface für Joystick und Maus  
- optimale Impulsfolge für jedes Game einstellbar  
- Dauerfeuer wird über Feuer- bzw. Maustaste aktiviert  
- einfach zwischen Maus/Joystick und Rechner stecken  
- abschaltbar

### Trackdisplay A2000 intern DF0: & DF1:

DM 98,-  
(neuer Power-LED Träger mit integrierten 7-Segmentanzeigen)

### Trackdisplay extern DF0: bis DF3:

DM 79,-  
für jedes Laufwerk einstellbar

### Drive-Expander

DM 39,-  
- für externe Laufwerke ohne Busdurchführung  
- einstellbare Laufwerksnummer  
- keine Kabellängenprobleme  
- abschaltbar  
- bei Verwendung eines Boot-Selectors kann von jedem  
Laufwerk gebootet werden

### BOOT-Selector für Amigas

DM 14,50  
wahlweise Booten von DF0: oder DF1: oder DF2: oder DF3:  
bei Bestellung bitte angeben

### BOOT-Selector elektronisch

DM 49,-  
wahlweise Booten von DF0: bis DF3:

### BTX/VTX Decoder mit FTZ (Drews)

DM 248,-  
Commodore BTX mit FTZ DM 179,-

### Midi-Interface

DM 89,-  
In/Thru/2\*Out, im Metallgehäuse für A500/A2000

### Staubschutzhäuben

AMIGA 500 DM 16,50  
AMIGA 2000 Keyboard DM 16,50  
Monitor 14" DM 29,50  
Drucker 10" DM 24,50  
Drucker 15" DM 29,50  
Die Staubschutzhäuben sind aus Kunstleder mit weichem  
antistatischem Innenfutter

### Software

RAM-Test Amiga DM 24,50  
zeigt defekte Speicherstellen grafisch an, 100% Assembler  
XCOPY II DM 49,-  
XCOPY II mit Hardwarezusatz DM 69,-  
TurboPrint II DM 89,-  
TurboPrint professional DM 189,-  
Quarterback (Festplatten-Backup) DM 119,-  
DPaint III DM 248,-  
Beckertext DM 189,-

### AMIGA-Computer

Amiga 2000 DM 1898,-  
Amiga 500 DM 928,-  
Amiga 500 BTX DM 998,-  
Harddisk A590 20MB für A500 DM 998,-  
Colormonitor Commodore 1084P DM 598,-

### Professional Drive Diskettenlaufwerke

\*\*\*\* Top in Qualität, Funktion und Design \*\*\*\*  
3 1/2"-Laufwerk AMIGA 2000 intern DM 159,-  
komplett mit Einbaueinheit und Anleitung  
3 1/2"-Laufwerk für alle AMIGAs extern DM 229,-  
abschaltbar, Busdurchführung, AMIGAFarben  
51/4"-Laufwerk für alle AMIGAs extern DM 279,-  
abschaltbar, Busdurchführung, 40/80 Tracks, AMIGAFarben

### Professional RAM-Board II A500 auf 1 MB

DM 199,-  
- superschnelle Megabit-RAMs (4\*511000)  
- mit Uhr & Datum  
- hard- und softwaremäßig abschaltbar  
- Superniedriger Stromverbrauch  
dto. Platine mit Uhr & Schalter ohne RAMs DM 79,-

### Professional RAM-Board III A500 auf 2,3 MB

DM 698,-  
- intern, inkl. Garry-Adapter  
- superschnelle Megabit-RAMs (16\*511000)  
- mit Uhr & Datum  
- Hard- und Softwaremäßig abschaltbar  
dto. Platine mit Uhr & Schalter ohne RAMs DM 198,-

### Professional RAM-Board A2000

DM 848,-  
8 MB mit 2 MB bestückt, autokonfigurierend, einfach durch  
Umstecken zusätzlicher RAMs und Jumper aufrüstbar, keine  
neuen PALs erforderlich  
dto. Platine teilbestückt ohne RAMs DM 498,-  
RAM-Satz für 2 Megabyte DM 498,-

### Genlock-Interface

DM 248,-  
nur Computer/nur Video/beide Bilder gemischt  
durchgeführter Monitorport, externes Gehäuse  
mit S-VHS und Cinch Buchsen

### Professional View

DM 998,-  
Video-Digitizer der neuen Generation  
On Board RGB-Splitter bereits integriert !!  
Genlock-Interface im Lieferumfang enthalten !!  
Realtime-Digitalisierung in 16 Graustufen, 25 Bilder/s  
HAM-Modus mit 4096 Farben,  
kurze Realtime Animationen in s/w, Übertragung 3,5MB/s !  
Alle Bilder und Animationen auf Diskette speicherbar,  
für Animationen mind. 1MB RAM  
unterstützt 1MB ChipRAM;  
Demo gegen frankierten Rückumschlag (DM 2,40) mit LeerDisk.

Haben Sie Hard- oder Software für den Amiga entwickelt ?  
Wir bieten Ihnen eine großzügige Umsatzprovision und eine  
ehrliche Abrechnung.  
>>>> Sprechen Sie uns an <<<<<<

Wir reparieren Ihren Amiga und Zubehör schnell und preis-  
günstig.

UPS-Versand: Nachnahme + 10 DM, Vorkasse + 5 DM,  
Großgeräte nach UPS-Tabelle ohne Aufschlag, Ausland nur  
gegen Vorkasse + 15 DM.  
Bei Vorkasse nur Eurochecks bis 400,- DM oder Über-  
weisung.  
Fordern Sie unser kostenloses Info an.  
Händleranfragen erwünscht!

## HK-Computer

F.Hansmann & Th.Küpper GbR  
Bonner Straße 37 \* 5000 Köln 1

Telefon: 0221/311606 \*\* Telefax: 0221/321166

Mo-Fr. 10.00-13.30 u. 14.30-18.30, Sa. 10.00-14.00

BTX: 0221321166

Stadtparkasse Köln, Kto. 6342133, BLZ 37050198



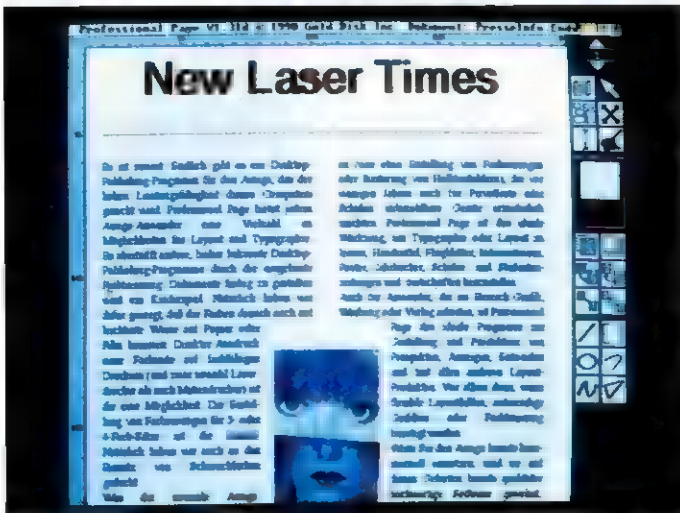


Bild 1. Professional Page ist das Layout-Programm, welches Texte und Bilder zu fertigen Layout-Seiten verknüpft.



Bild 2. Professional Draw heißt das Zeichenprogramm, das voll auf den DTP-Betrieb zugeschnitten ist.

Durch Umwandeln der ursprünglichen Textfiles in das PostScript-Format.

Ein kleines PostScript-Beispiel-Programm zeigt Abbildung 1. Das Ergebnis, also das, was der Drucker unter Zuhilfenahme des PostScript-Interpreters daraus macht, können Sie in Abbildung 2 sehen. Der Aufbau der kleinen

Datei entspricht fast der von bekannten Programmiersprachen, als Vergleich könnte am besten die Sprache 'FORTH' dienen. Das Besondere an der ganzen Sache ist, daß PostScript-Dateien mit einem beliebigen Editor eingegeben werden können; als einfachstes Beispiel kann hier der ED aus dem C-Verzeichnis stehen.

Nur eines ist wichtig, die PostScript-Dateien müssen reinen ASCII-Code enthalten, nur dann werden sie vom Interpreter auch ohne Fehler übersetzt. Natürlich ist es nicht Sinn einer solchen Anlage, vom Anwender zu verlangen, daß er sich erst mal mit PostScript auseinandersetzt, auch wenn dadurch der Nutzen des DTP noch steigt. Deshalb liegt dem Paket auch gleich eine passende Software bei: Professional Page und Professional Draw.

neal (in verschiedenen Maßen darstellbar) auf den Bildschirm gebracht. Man kann diese Seite in allen Variationen betrachten, also Teilausschnitte der Seite oder Vergrößerungen oder Verkleinerungen.

Die angelegte Seite wird nun in Boxen unterteilt, wobei jede Box unterschiedlich gefüllt werden kann – entweder mit Text oder mit Grafik. Da PPage im hochauflösenden Modus arbeitet, lassen sich auch dementsprechende Bilder einlesen, seien sie digitalisiert oder mit Malprogrammen erstellt. In unserem Fall diente ein Demo-Bild des Malprogrammes 'Photon Paint' von MicroIllusions als Beispiel. Das Bild wurde komplett eingelesen und an die Größe der Box angepaßt. Der Text entstammt einem AMIGA-DOS-Beitrag und wurde als ASCII-Text eingelesen.

In puncto Bilder möchte man irgendwann selbst seine kreativen Fähigkeiten einsetzen und seinem Zeichentalent freien Lauf lassen. Dann bietet sich Programm zwei an: Professional Draw. Professional Draw ist ein Zeichenprogramm, das für die Arbeit mit

## Vor dem Ausdruck steht die Erstellung – Der Anwender ist Redakteur, Layouter und Druckerei gleichzeitig

Professional Page ist das Layout-Programm, welches Texte und Bilder in die Form bringt, die schließlich per Laserdrucker ausgegeben werden soll. 'PPage' (so der abgekürzte Name) kann Texte verschiedener Art übernehmen. Darunter solche der bekanntesten Textverarbeitungsprogramme. Sinnvoller ist aber auf jeden Fall das Einlesen von reinen ASCII-Texten, die auf jedwede Steuerzeichen verzichten. PPage bietet genug Möglichkeiten, um den Text trotzdem individuell gestalten zu können.

Anfangen wird natürlich mit dem Aussehen der Seite. Bei unserem Test beließen wir es bei einer Standard-DIN A4-Seite, die als Testobjekt auf jeden Fall ausreicht. Diese Seite wird mit passendem Li-



Abb. 2. ...und interpretiert vom PostScript-Interpreter.

```

■ PostScript-Testausdruck
% für Apple-Laserwriter-Plus-kompatible
■ PostScript-Emulationen

/Rechteck      ■ Prozedur definieren
{
    1 index 0 rlineto
    ■ zweiten Parameter vom Stack holen
    0 exch rlineto
    ■ ersten Parameter interpretieren
    neg 0 rlineto
    ■ Rechteck fertigzeichnen
    closepath
} def

% Hauptprogramm

/ZapfChancery-MediumItalic findfont
% Schrift wählen
36 scalefont % Schriftgröße 36p.
setfont % setze aktuellen Font
newpath % neuen Grafikweg öffnen
0.0 setgray % aktuelle Farbe: Schwarz
170 460 moveto % "Stift" positionieren
175 120 Rechteck

fill % Aufruf der Prozedur
stroke % Fläche füllen
1.0 setgray % Grafikweg ausführen
215 535 moveto % aktuelle Farbe: reines Weiß
(Lies) show % "Stift" positionieren
185 505 moveto % erste Textzeile schreiben
(AMIGA) show % "Stift" positionieren
270 480 moveto % zweite Textzeile schreiben
(DOS) show % dritte Textzeile schreiben
showpage % Ausgabe der Seite auf Drucker

28 scalefont % neue Schrift wählen
setfont % Schriftgröße 28p.
0.7 setgray % setze aktuellen Font
270 480 moveto % 30 Prozent Grauraster
(DOS) show % "Stift" positionieren
showpage % dritte Textzeile schreiben
showpage % Ausgabe der Seite auf Drucker
    
```

Abb. 1. Ein kleines PostScript-Programm, erstellt mit einem ASCII-Editor...



# ABONNEMENT



INFORMATIONEN AUS  
ERSTER HAND

**Ein Abonnement ist  
praktisch und bequem.**

**Widerrufsrecht**

Jeder Abonnent hat das Recht, seine Bestellung innerhalb einer Woche beim DMV-Verlag, Postfach 250, 3440 Eschwege, schriftlich zu widerrufen. Die rechtzeitige Absendung des Widerrufschreibens genügt zur Fristwahrung.

**AMIGA DOS  
kostet im Abonnement:**

Im Inland bzw. West-Berlin:

6 Ausg. = 35,- DM

12 Ausg. = 70,- DM

Im europäischen Ausland:

6 Ausg. = 50,- DM

12 Ausg. = 100,- DM

Im außereuropäischen Ausland:

6 Ausg. = 60,- DM

12 Ausg. = 120,- DM

Bitte benutzen Sie die Bestellkarte.

**DMV-Verlag · Postfach 250 · 3440 Eschwege**





DTP konzipiert worden ist, PDraw ist nämlich objekt-orientiert. Während man bei Deluxe Paint quasi auf einer Leinwand zeichnet und diese auch nur komplett wieder säubern kann, ist es bei PDraw möglich, alle Elemente eines Bildes als Objekte an-

zusehen, die man nach Belieben verschieben, ersetzen, löschen oder übereinanderlegen kann. Jedes Objekt behält seinen Status, auch wenn das komplette Bild aus lauter Objekten zusammengefügt wurde. Und damit ist auch klar, warum PDraw der DTP-Anlage

ge beiliegt: Mit PDraw lassen sich schnell Gestaltungselemente für DTP-Seiten erstellen.

Bei beiden Programmen ist allerdings eine gewisse Einarbeitungszeit vonnöten, so komplex sind sämtliche Funktionen.

Sind Grafik und Text bereit zur Übernahme, werden sie in die entsprechenden Boxen gesetzt. Hier nun kann die Feinarbeit beginnen. Einzelne Zeilen können zu Überschriften umgewandelt werden, Schriftarten lassen sich im Text anwählen, ganze Textpassagen können ein gänzlich anderes Aussehen bekommen.

Alles in allem ist jede Komponente des DTP-Pakets ein Erlebnis für sich; alles zusammen bildet eine Einheit, die es jedem ermöglicht, mit DTP praktische Erfahrungen zu sammeln und DTP konkret anzuwenden.

### **“Wer soll das bezahlen?” – Gedanken über eine mögliche Käufergruppe**

Daß die DTP-Anlage nichts für den kleinen Geldbeutel ist, dürfte jedermann verständlich sein. Amiga 2500 mit 68020-Karte und eingebauter Festplatte sowie ein NEC 'Silentwriter LC890' (der allein schon über 12000,- DM kostet) plus Professional Page und Professional Draw für knapp 30000,- DM, da wird wohl kaum der 'Kleinanwender' in Frage kommen. Dies wäre so, als würde Handwerksmeister Schmidt für seine Fahrt zur Arbeitsstätte einen Jumbo-Jet benutzen.

Die eigentlichen Käufer sollen, wie schon gesagt, Menschen sein, die mit dem Erzeugnis 'DTP-Anlage' produktiv arbeiten. Produktiv heißt zum Beispiel Einsatz in der Werbebranche oder zu Präsentationszwecken. Und hier sind ein paar Punkte negativ anzukreiden:

Professional Page und Professional Draw mögen zwar auf dem Amiga sehr gut sein, der Amiga selbst macht aber in Punkto DTP dem PC noch keine große Konkurrenz. Gerade beim PC gibt es inzwischen eine erkleckliche Anzahl von sehr guten DTP-Programmen, mit denen Professional Page nicht so richtig mithinkt. Die Geschwindigkeit des Programms ist nicht sehr hoch, das dauernde Wechseln zwischen Menüleiste und Mauszeiger kann auf die Dauer ermüden. Auch beim Punkt Bildschirmdarstellung ist einiges zu bemängeln. PPage arbeitet im Interlace-Modus,

## Der Desktop Publisher "Professional Page" im Test

### Malprogramm als Sound-Editor

Wer hätte gedacht, daß man mit einem Malprogramm Sounds kreieren oder bearbeiten kann? Dieses Utility macht's möglich! SOUND -welch ein phantastisches Schlagwort! Die einen denken dabei sofort an den Hörgenuß. Nur einige wenige denken dabei auch an Wellen, Frequenzen, Hüllkurven und Amplituden. Doch gerade für diese Anwender ist dieses Programm von Nutzen, denn es bietet die Möglichkeit, kreativ eine Hüllkurve nach individuellem Geschmack zu konstruieren oder zu editieren.

Alles, was man benötigt, ist Dpaint3 oder ein ehrliches Malprogramm, welches nicht im Hold-And-Modify-Modus arbeitet, und schon kann's losgehen. Zuerst müssen zwei CLI-Prozesse aktiviert sein. Dies geschieht über den NEWCLI-Befehl, wenn bereits ein CLI-Prozeß über die Workbench oder ein SHELL aktiv ist. Von dem einen CLI-Fenster ruft man Wave.obj auf, von dem anderen danach DPaint3, indem man den Screen nach unten zieht oder mit <Amiga links + ESC> in den Hintergrund legt. Ist dies geschehen, so wählt man im



Setup ein Screen-Format von 640\*256 Punkten mit 2 Farben. Die Arbeit mit dem Malprogramm verläuft wie gewohnt.

Als erstes sollte man im "Prefs"-Menü "Coords" wählen um immer über die graue Mausibraut Bescheid zu wissen. Wenn man berücksichtigt, daß die x-Achse für die Hüllkurve durch die Bildschirmpunkte (0,127) und (640,127), also waagrecht in der Bildschirm-Mitte verläuft, wird man feststellen, daß die Amplitude zwischen -128 (=0) und +128 (=255) liegen kann. Mit Sicherheit ist das kein Zufall, denn ein Sample besteht aus einer Folge von vorzeichenbehafteten 8-Bit-Werten. Da der Screen nun 640 Pixel breit ist, lassen sich auf ihm genau 640 Byte eines Samples darstellen. Nehmen wir einmal an, eine entsprechende Hüllkurve sei gezeichnet. Jetzt müssen wir mit F-10 die Menüleiste abschalten, damit sie

Abb. 3. Eine Testseite, erstellt mit Professional Page und mit Hilfe eines Demo-Bildes von PhotonPaint



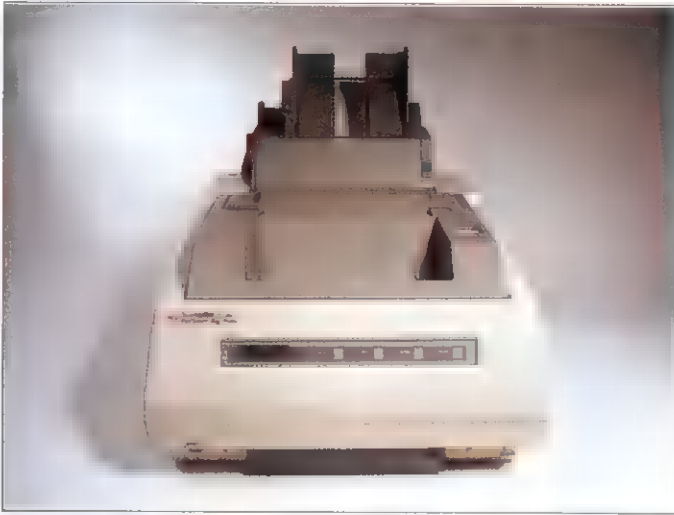


Bild 3. Der NEC Silentwriter LC890, ein Laserdrucker der Sonderklasse. Grafik und Text in verschiedenen Schriftarten sind kein Problem, PostScript-fähig ist er sowieso

der zwar dank dem neuen Grafik-Chip und dem Multisync-Monitor nicht mehr das flackernde Bild liefert wie sonst, trotzdem werden die Augen beim Arbeiten ganz

schön beansprucht. Auch das Positionieren innerhalb der Boxen ist recht mühsam, da die Amiga-Maus nicht gerade feinfühlig zu nennen ist. Das Zusammenspiel PDraw und

PPage klappt reibungslos, der Amiga brachte es allerdings zustande, nach Fertigstellung einer Seite vor Ihrem Ausdruck mit einem 'Guru' auszusteuern. Zufall? Man kann es nicht genau sagen, jedoch ein unschöner Umstand für diejenigen, die sich die DTP-Anlage um des DTP willen gekauft haben und nicht, um die Exceptions des Betriebssystems kennenzulernen.

Den Amiga als Desktop-Publishing-Gerät zu sehen, ist eine neue Erfahrung. Ob er allerdings seinen stärksten Konkurrenten, den PCs mit ihrer jetzt schon starken Software, Paroli bieten kann, ist fraglich. Dazu müßten Soft- und Hardware noch genauer aufeinander abgestimmt werden – und eine größere Stabilität erreichen.

Trotz allem: Die Ergebnisse, die schon nach kurzer Einarbeitung zu erzielen sind, sind sehenswert...

(Peter Schmitz/jb)

## AMIGA DOS Info

### AMIGA 2500 DTP

Das Amiga Desktop Publishing System basiert auf einem A 2500 mit 68020-Karte und 3 MByte RAM-Speicher. Fest eingebaut ist eine 44 MByte AutoBoot-Festplatte unter Kickstart und Workbench 1.3. Ausgeliefert wird die DTP-Anlage in mehreren Variationen, darunter mit 68030-Prozessor und Thermotransfer-Drucker; die uns zur Verfügung stehende wurde durch einen NEC-Laserdrucker 'Silentwriter LC890' mit eingebautem PostScript-Interpreter ergänzt. Beiliegende Programme sind: Professional Draw und Professional Page. Preis der Anlage: zirka DM 30000, -

Einsatzgebiet: Professionelle Layout-Erstellung.

## NORDSOFT

### PUBLIC DOMAIN

#### \*\*\* 4000 Disketten \*\*\*

Einzelstück	je 2,90 DM
ab 10 Stück	je 2,70 DM
ab 20 Stück	je 2,50 DM
ab 50 Stück	je 2,30 DM
ab 100 Stück	je 2,10 DM

2 Katalogdisketten gegen 5,- DM anfordern!

### LAUFWERKE

NORDSOFT 3,5" NEC 1037A Bus  
abschaltbar, extern ■ 229,-  
dto. mit Trackdisplay. ■ 299,-

### SPEICHERKARTEN

Speicher für A500-2000?  
Wir haben die aktuellen Preise!!  
SRAMS/DRAMS? Leerplatinen?  
Erfragen Sie die Tagespreise!!

### ZUBEHÖR

Big Agnus-8372A DM 159,-  
Kick-Rom 1.3 DM 59,-  
Diverse Erweiterungskarten  
für PC/AT-Karten lieferbar!! z.B.  
Umbau XT-Karte ■ MHz DM 198,-  
Festplatten/Filecards mit A.L.F.V.2.0  
NORDSOFT! DER AMIGA CENTER!!  
Geld sparen? Klar! Festplatten = Selbstbau! Rufen Sie durch oder besuchen Sie uns!

Modem Supra 2400 DM 389,-  
Modem A2000 int. DM 398,-

### ANWENDER

Bitte fordern Sie unseren SOFTWARE-KATALOG für Ihren Rechner typ an! (AMIGA/PC/Atari etc.)

### GAMES

Günstige Sonderpreise und aktuelle Neuerscheinungen!! Informieren Sie sich auch über Gebrauchtspiele!!

Allgemeine Lieferbedingungen: Versand nur per Nachnahme oder Vorkasse. Bei Nachnahme Versand und Verpackungskosten DM 10,-; Vorkasse DM 8,-; Auslandsbestellung nur Vorkasse. Lieferzeit: Innerhalb 1 Woche sofern am Lager.

NEU! Ladenlokal Bremen, Heidbergstraße 75, TELEFON: 0421/61 1430

Schweneker & Behnke

28 Bremen 21 • Rostocker Str. 52

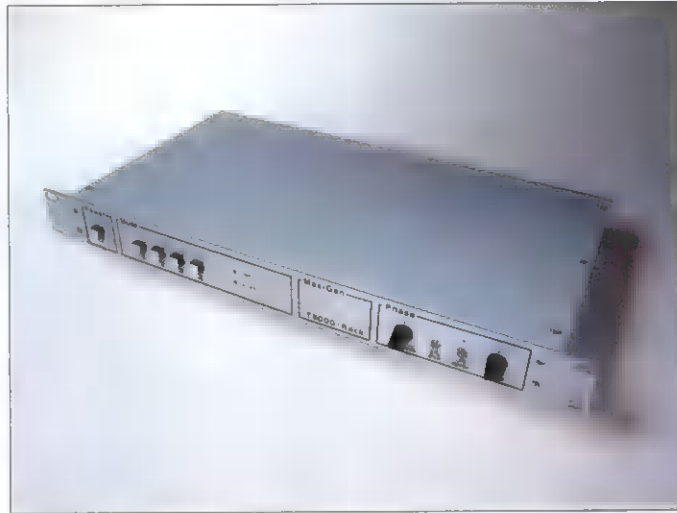
<p><b>AMIGA -- BTX MULTITERM</b> und Multiterm PRO</p> <p><b>Preise:</b> Multiterm an A-Koppler / Modem 125,-DM Multiterm Pro 153,-DM Multiterm Post-Modem DBT03 195,-DM Multiterm Pro 233,-DM Sep. Interface Amiga-DBT-03 auf Anfrage</p> <p><b>Das richtige Modem ....</b> Best 2400 Plus (Btx-fähig)* 429,-DM Discovery 2400C* 359,-DM *Modems sind ohne ZZf (Betrieb am Postnetz der BRD und Westberlin ist unter Strafe gestellt.) 512K mit Uhr (A500) 189,-DM Externes LW 3,5" 219,-DM Info über MS-DOS Rechner und weitere Neuheiten anfordern.</p>	<p><b>Kopierschutz passé, hier ist Project D.</b> Über 300 kopiergeschützt Programme kopierbar. Das Original direkt ■ USA. Incl. deutscher Anleitung ■ 85,-DM</p> <p><b>AMIGA SOFTWARE HIGHLIGHTS</b> DIGI-View 4.0 298,-DM Pagestream 1.8 339,-DM Pagesetter II 169,-DM</p> <p><b>Die MS-DOS Welt ....</b> Flugsimulator 4.0 Die neueste Version von Microsoft. Direkt aus USA. 139,-DM Text III Layout. Textverarbeitung mit Ausdruck bis zu 360x360 Punkten. (EGA/Hercules) Mit 8 Schriften nur 378,-DM 25 weitere Schriften 90,-DM</p>	<p>**** Preiswerte Software ab 2,40 DM ****</p> <p><b>PD-Schnell...Versand</b> Wir liefern: KICKSTART, FISH, TAIFUN, RPD, AUG, CACTUS, TBAG, PANORAMA, RUHR, usw. Spielpakete I, II, III Je 10 Disketten mit 26-43 Spielen, teilw. mit deutschen Anleitungen. Je 49,-DM Einstieger III 10 Disketten, die den Einstieg erleichtern. Spiele, Erklärungen, Utilities, etc. 39,-DM Das goldene PD-Buch Incl. 10 Disketten 105,-DM Alle PD-Bücher Incl. 42 Disk 325,-DM 2 Katalogdisketten (Bar, Briefmarken) 5,-DM Ausführliche Disk-Beschreibungen und Infos, ca. 4000 Disk im Bestand !!!</p>	<p><b>Software Komplett !</b> Haushaltsbuch, Kontoführung, Textverarbeitung, mCAD, Anti-Virus-Disk, Spiele, Vokabeltrainer, Schallplatten-/Videoverwaltung, Datenbank, Adressverwaltung, Schach, Utilities. 20 TOP-PD-DISKETTEN, die keinen Wunsch offenlassen. Viele deutsche Anleitungen. 79,-DM Money-Player Geldspielgerät. Wie in der Spielhalle. Das Original !! 39,-DM Danger Castle Unheimliches Erlebnis, Supergrafik, und -sound. 39,-DM Soccer Manager Plus Der Fußball-Manager 49,-DM</p>
<p><b>M. Kirschbaum Medienberatung</b> Schubertstr. 3, 4320 Hattingen Tel. + Btx: 02324/82249 — Fax: 02324/83722</p>		<p><b>Hard- und Software SCHOLLE</b> Pilgrimstr. 6, 4630 Bochum Tel.: bis 21 Uhr 0234/ 770388 Fax: 0234 / 73867</p>	



**W**enn man sich eine Übersicht über die zur Zeit erhältlichen Genlocks verschafft, stellt man unweigerlich fest, daß doch große Unterschiede hinsichtlich der Bildqualität bestehen. Da vermag so manches Produkt nicht zu halten, was die aufwendig gestylte Verpackung verspricht. Ganz anders jedoch das Maxi-Gen von Merkens EDV, das mit Sicherheit zu den besten Systemen seiner Klasse gehört, so viel schon einmal vorweg. Der Lieferumfang besteht aus dem eigentlichen Genlock Interface, das in einem soliden 19-Zoll-Einschubrahmen montiert ist, sowie einem Verbindungskabel, mit dessen Hilfe das Genlock zwischen den Amiga und den Monitor geschaltet wird.

Ein externes Netzteil versorgt das Genlock mit Spannung, so daß das Amiga-Netzteil nicht unnötig belastet wird. Ein Handbuch klärt die wichtigsten Fragen zur Inbetriebnahme und erläutert die Anschlüsse, die an der Rückseite des Gerätes montiert ist. Der Anschluß an den Amiga, in unserem Fall fand ein A 2500 Verwendung, ist mit Hilfe des beiliegenden Kabels Momentsache. Platz findet das Genlock wahlweise in einem 19-Zoll-Rack, oder einfach auf der Zentraleinheit.

Auffallend sind einige Anschlüsse auf der Rückseite, die aufzeigen, daß das Maxi-Gen für den professionellen Bereich konzipiert wurde. Speziell die Anschlüsse <Ext Control>, <Ext Coder>, <Ref IP> und <Blk IP> zeigen auf, daß hier ein ausbau-



## Profi-Genlock

**Grafik und Videoverarbeitung sind eines der interessantesten und vielseitigsten Themen auf dem Amiga. Und waren Genlocks, die in professioneller Qualität arbeiten, bis vor einem Jahr für den Heimanwender noch unerschwinglich, hat sich diese Situation sehr zum Vorteil des Heimanwenders geändert. Begleiten Sie uns doch einfach auf unserem Streifzug durch die Welt der Bilder.**

fähiges Grundgerät vorliegt, das mit entsprechender Peripherie zu einem kleinen, aber leistungsfähigen Fernsehstudio ausgebaut werden kann.

Eine genauere Ausführung der Möglichkeiten, die die eben genannten Anschlüsse bieten, würde sicherlich den Rahmen dieses Berichts sprengen, wir werden an anderer Stelle in einem Grundlagenartikel auf diese Thematik eingehen.

Um sinnvoll arbeiten zu können, ist darüber hinaus eine externe Videoquelle erforderlich, die mittels eines BNC-Anschlusses in das Genlock eingespeist wird. Hier kann eine Videokamera oder auch ein Videorekorder benutzt werden. Wollen Sie die Ergebnisse Ihrer Arbeit verewigen, so benötigen Sie einen zweiten Videorekorder, um Eingangssignal und Computergrafik zusammen auf ei-

nem Medium aufzuzeichnen. Soweit zu den Vorbereitungen, doch was läßt sich mit dem Genlock praktisch durchführen?

## Jetzt kommt Bewegung ins Spiel

Neben praktischen Anwendungen wie das Versehen selbstgedrehter Videofilme mit Titel und sonstigen Credits, kommt das Genlock in dem Moment zur Geltung, wo eine Mischung mit Elementen eines Grafikprogramms erfolgt. Eine mit dPaint erzeugte Animation läßt sich beispielsweise problemlos in das laufende Videobild setzen. Damit sind Effekt- und Verfremdungsmöglichkeiten gegeben, die die Fähigkeiten

### Technische Daten

#### Netzteil:

Primär: 220/240 Volt, 50Hz;  
Sekundär: 15 Volt, 12 VA

#### Eingänge:

CVBS: 1V/ss  
REF: 0,3 V Sync  
0,3 V: Burst  
BLK: 2V/ss

#### Ausgänge:

RGB: 0,7 V/ss  
Sync: 0,3 V/ss  
CVBS: 1,0 V/ss

#### Ext Coder:

RGB: 0,7 V/ss  
Sync: 0,3 V/ss  
Key: 0,3 V/ss Sync + 0,7 V/ss Key

Alle Anschlüsse weisen eine Impedanz von 75  $\Omega$  auf.

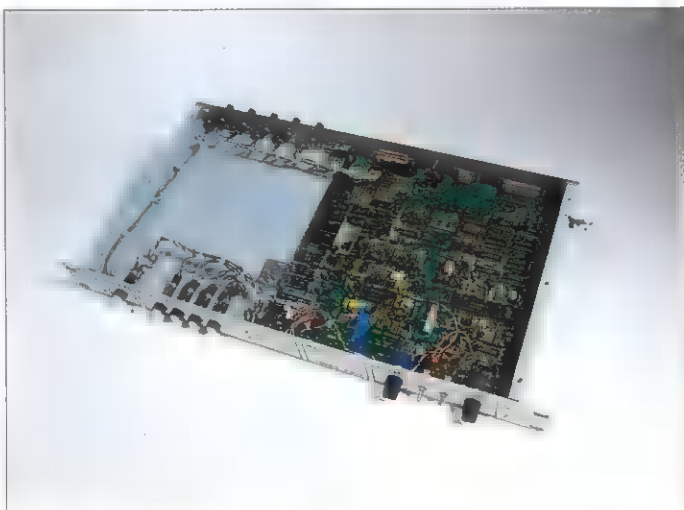


Bild 1. Das Innenleben des Maxi-Gen von Merkens EDV zeichnet sich durch übersichtliche, solide Verarbeitung ■■

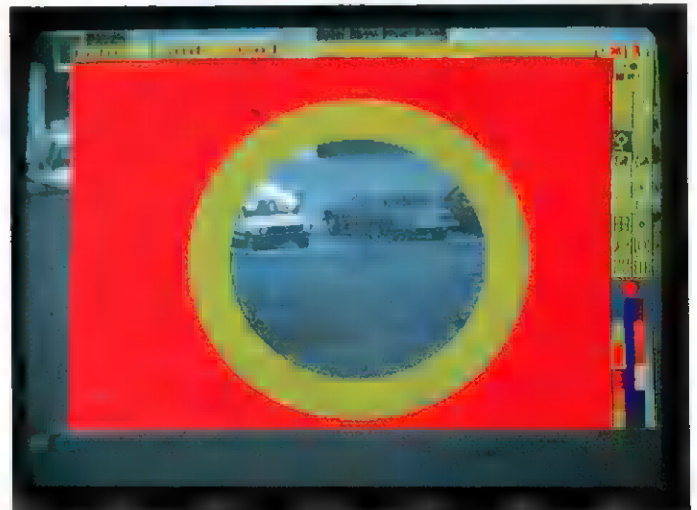


Bild 2. Die Kombinationsmöglichkeiten des Genlocks sind,...



des Amiga voll zur Geltung kommen lassen. Dabei ist es gleichgültig, in welcher Auflösung amigaseitig gearbeitet wird. HiRes wird ebenso pro-

blemlos integriert, wie Interlace oder eventuelle Over-scan-Modes.

Praktisch sind dabei die "Mixschalter", mit denen festgelegt

## Glossar

### Doch was genau ist eigentlich ein Genlock-Interface? Wo liegen Unterschiede zu den ebenfalls populären RGB-Splittern bzw. Digitizern?

Keine Angst, hier sollen jetzt nicht theoretische Grundlagen besprochen werden. Ein kleiner Exkurs ist jedoch notwendig, um eventuelle Mißverständnisse im Keim zu ersticken.

#### RGB-Splitter

Da wäre zunächst der RGB-Splitter. Dieses Gerät hat die Aufgabe, ein Videosignal in seine einzelnen Farbkomponenten aufzusplitten: den ROT-, GRÜN- und BLAU-Anteil. Ergänzt werden diese drei Auszüge durch ein Synchronisations-Signal, um eine phasenrichtige Zuordnung der einzelnen Auszüge zu gewährleisten. Vorteil: Verarbeitet man die einzelnen Komponenten anschließend weiter, so sind eine höhere Farbreinheit und Detailtreue als bei der direkten Verarbeitung des Videosignals gewährleistet.

Ursache: Beim Videosignal werden alle Informationen einem HF-Träger aufmoduliert. Zur Rückgewinnung der Information muß das Signalgemisch demoduliert und beispielsweise in einer Matrix decodiert werden.

Das Signal, das letztendlich am Ausgang des RGB-Splitters abgegriffen werden kann, ist allerdings vom Amiga noch nicht direkt zu verarbeiten.

Verwendung findet der RGB-Splitter z.B. bei Standbild-Digitizern, um auf die Verwendung einer Farbfilterscheibe verzichten zu können.

#### Die Digitizer

Ein Digitizer dagegen hat die Aufgabe, ein ankommendes Video- bzw. HF-Signal in für den Computer verwertbare Spannungsspiegel umzusetzen. Dies wird in der Regel durch einen AD-Wandler erreicht und an der parallelen Schnittstelle in den Amiga eingespeist. Mit Hilfe entsprechender Software können nun die einzelnen Auszüge im Amiga wieder zu einem Gesamtbild zusammengesetzt und weiterverarbeitet werden.

Grundsätzlich muß bei Digitizern zwischen Realtime-Applikationen und Standbild-Digitizern unterschieden werden.

#### Genlock-Interfaces

Ein Genlock-Interface dagegen dient der Kombination zweier Bildquellen, in unserem Fall also der Überlagerung eines Videobildes und einer Computergrafik. Hört sich zunächst einfach an, gestaltet sich von der technischen Seite jedoch schon etwas anspruchsvoller, da sich die beiden zu vermischenden Signale weder in Phase, Amplitude noch Frequenz entsprechen. Auch mit der Synchronität ist es so eine Sache. Wenn beide Signalquellen nicht einen einheitlichen Takt aufweisen, muß zwangsläufig ein Signal "durchlaufen". Ein Effekt, der aus verstimmten Fernsehern ja hinlänglich bekannt ist. Zudem muß differenziert werden, an welchen Stellen sich die Signale überlagern und an welchen Stellen nicht. Schließlich sollen beide Bildquellen klar erkennbar sein. Transparenz heißt hier das Zauberwort, was nichts weiter bedeutet, als daß eine Farbe des Computerscreens als transparent oder Hintergrundfarbe erklärt wird. In der gemeinsamen Darstellung wird dann die Transparentfarbe zugunsten des eingespeisten Videobildes unterdrückt. Der Effekt: Alle Stellen der Computergrafik, die in der Hintergrundfarbe dargestellt werden, entfallen und werden durch das Videobild ersetzt. Als Beispiel mag hier der Vor- bzw. Abspann eines Filmes dienen, in dem einerseits das letzte Bild des Films, andererseits aber auch die Namen der Darsteller eingeblendet sind.

Soweit zur Begriffsentwerrung im Hardware-Dschungel.



Bild 3. ...wie man sieht, mehr als vielseitig, und für Präsentationen bestens geeignet

werden kann, in welcher Weise die Bildquellen verknüpft werden. Da wären die Zustände <nur Video>, <nur Amigagrafik>, <Amigagrafik über Videobild> und <Stanzen auf Farbe 0>.

Ergo kann praktisch zwischen dem Videoscreen und dem Grafiksreen umgeschaltet werden, was sowohl die Arbeit an der Grafik als auch die Kombinationen ungemein erleichtert. Stehen Videosequenz, Grafik und Animation, kann problemlos das Ergebnis aufgezeichnet und beispielsweise zu Präsentationszwecken gut verwendet werden. Dabei darf ein Umstand nicht übersehen werden. Das Signal, das die Videoquelle zur Verfügung stellt, kann nicht verändert werden, es ist also in dieser Konfiguration unmöglich, auf das Videobild amigaseitig zuzugreifen, da dieses Signal im Amiga ja nicht zur Bearbeitung zur Verfügung steht. Entsprechend muß also die Motivwahl für die Videoquelle erfolgen. Außerdem sollte darauf geachtet werden, die Motivgröße so zu wählen, daß eine sinnvolle Verknüpfung mit Grafikelementen noch möglich ist. Ein Firmenlogo auf der Stirn des Firmensprechers ist zwar sicherlich lustig, entspricht wahrscheinlich aber nicht den Vorstellungen über das fertige Produkt. Apropos Produkt: Die letztendlich als Mischung aufgezeichnete Kombination hat Fernsehqualität und braucht sich hinter den Ergebnissen manch professioneller Studios nicht zu verstecken, wobei das Endprodukt zum größten Teil von der

Qualität der eingespeisten Signale abhängt.

Ob Präsentation, Bearbeitung von Urlaubsfilmen oder die Erzeugung eines Videoclips, die Möglichkeiten des Genlocks sind so vielseitig wie die Phantasie des Benutzers.

(mm)

## AMIGA DOS Blitzlicht

**Name:** Maxi-Gen

**Hersteller:** Merkens EDV Computer-Video-Systeme Fuchstanzstr. 6A, 6231 Schwalbach, Tel.: (06196)-3026

**Preis:** 2789,- DM

**Besonderheiten:** Kann in Zusammenarbeit mit einem Mischer mit Superimpose-Fähigkeiten extern gesteuert werden.

#### Positiv:

- reichhaltige Anschlußmöglichkeiten
- professionelle Bildqualität
- solide 19-Zoll-Einschubtechnik
- externe Spannungsversorgung
- unkomplizierte Bedienbarkeit
- arbeitet mit allen Kameras gut zusammen
- einfacher, problemloser Aufbau
- sinnvolles, funktionales Bedienfeld
- gutes Preis/Leistungsverhältnis

#### Negativ:

- kein Hauptschalter
- Kanten des Einbaurahmens nicht entgratet



**D**er Amiga ist ein Computer mit leistungsstarken Soundfähigkeiten. Er ist in der Lage, digitalisierte Töne abzuspielen, ohne den Hauptprozessor zu belasten. Weiterhin kann er vier Klänge gleichzeitig auf zwei Kanälen wiedergeben. Doch ist es beim Amiga wie bei "normalen" Musikinstrumenten: Bevor man in den Genuß kommt, die hervorragenden Musiken zu Gehör zu bekommen, muß man in der Lage sein, das Instrument zu bedienen. Bei dem Amiga ist es da etwas einfacher, wenn man sich eines Sound-Editors bedient. Wir haben uns wieder einmal auf dem Softwaremarkt umgesehen und einen Sound-Editor unter die Lupe genommen, der die Erstellung von Musikstücken erleichtert.

Das TFMX-Soundtool ist ein Hilfswerkzeug zur Musikprogrammierung, das auf den ersten Blick durch seine Vielzahl an Optionen etwas unüberschaubar wirkt. Hat man sich jedoch erst einmal mit der Bedienung dieses Editors vertraut gemacht, entdeckt man Funktionen, von denen sich so manches Soundtool eine Scheibe abschneiden könnte. Allein schon die Tatsache, daß der "Komponist" über drei unterschiedliche Editoren (Pattern-, Track- und Makro-Editor) verfügt, macht diese Software zu einem individuellen Musik-Tool.

TFMX verarbeitet nur Samples, also digitalisierte Geräusche – die Amiga-internen Sounds werden ignoriert. Dies ist jedoch vom Klangtechnischen kein großer Nachteil, da die internen Töne des Amiga ohnehin für Musiker von untergeordneter Bedeutung sind.



Das TFMX-Soundtool wartet nicht nur mit hervorragenden Soundeffekten, sondern auch mit beispielhafter Bedienungs-freundlichkeit auf

## TFMX – Ein Soundgigant?

### Ein Soundtool nicht nur für Profis

Vielen Amiga- und C64-Besitzern dürfte der Name Chris Hülsbeck schon ein Begriff sein, hat er doch in diversen Computerspielen für die entsprechende musikalische Untermalung gesorgt. Diesmal zeichnet er für einen Soundeditor verantwortlich, der sich wirklich sehen lassen kann.

Lediglich der Speicherplatz wird durch die umfangreichen Samples beeinträchtigt.

Wobei wir auch schon bei der Speicherbelegung des TFMX wären. Der Benutzer wird nach dem Booten der Diskette aufgefordert, die entsprechen-

de Rechnerkonfiguration (512 kByte bzw. 1 MByte) anzugeben. Speichererweiterungen auf über 1 MByte werden zur Zeit noch nicht unterstützt.

Für Rechner mit einer Speicherkapazität von 512 kByte stehen 25 kByte Musikdaten zur Verfügung, während diejenigen, die einen Amiga 2000 bzw. eine Speichererweiterung besitzen, auf 50 kByte Musikdaten zugreifen können.

Dabei unterscheidet TFMX innerhalb dieser beiden Konfigurationen noch zwei weitere Darstellungsformen (16 und 4 Farben). Wählt man die Vierfarb-Darstellung, stehen den 512-kByte-Rechnern 225 kByte und den 1-MByte-Amigas 355 kByte Samplespeicher an, im 16-Farben-Modus sind es dagegen nur 145 bzw. 275 kByte. Die Erstellung eines Musikstückes fängt mit dem Einladen der Samples an. Nebenbei bemerkt: Das TFMX-Soundtool wird mit drei Disketten ausgeliefert, wobei auf zwei Disketten nur Sound-Samples im IFF-Format zu finden sind. TFMX akzeptiert verschiede-

ne Sampleformate wie IFF, DATA und Future Sound.

### Patterns im Viervierteltakt

Nachdem die "Instrumente" in Form von Samples (bis zu 256 Samples möglich) bereitgestellt sind, gilt es, den Takt einzustellen. Hier hilft eine kleine Besonderheit des Editors – das Metronom, also der Taktgeber, das wahlweise mit einer gerade abspielenden Melodie parallel laufen oder auch ausgeschaltet werden kann. Dieses Metronom ist frei einstellbar und verwendet nicht die Soundkanäle.

Nun stehen dem Benutzer zwei Möglichkeiten der Sounderzeugung zur Verfügung: zum einen das direkte Einspielen von Musikstücken per Klaviatur und zum anderen die Programmierung. Gehen wir zunächst den einfacheren (und schnelleren) Weg, nämlich den der Musikerzeugung per Klaviatur.

TFMX verfügt über eine Record-Page, womit man über die Tasten des Amiga Musikstücke spielen kann. Die gespielten Stücke können abgesichert und nachträglich bearbeitet werden. Hier stellt das Metronom ein nützliches Werkzeug zur Verfügung. Es lassen sich sogar Auftakte als Vorzähler vorgeben. Doch gehen wir lieber auf den interessanteren Teil des TFMX ein, dem Programmieren von Kompositionen.

Die Samples stehen ja schon zur Verfügung, fehlen eigentlich nur noch die Noten und die Toneffekte. Dazu sehen wir uns die drei Editoren etwas genauer an. Im Pattern-



Der Vierfarb-Editor – eine etwas unübersichtliche Darstellung, dafür jedoch mehr Platz für Samples

### AMIGA DOS Blitzlicht

**Name:** TFMX-Editor  
**Hersteller:** Demonware  
**Quelle:** Fachhandel  
**Preis:** zirka 130 DM

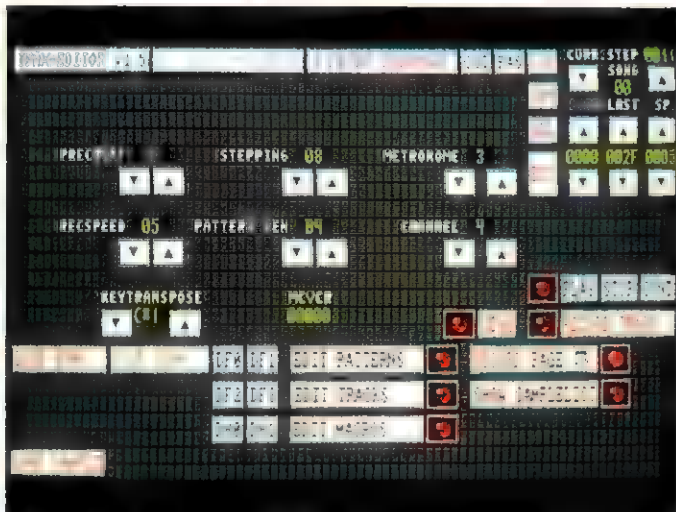
#### Positiv:

- unterschiedliche Sample-Formate einlesbar
- Musikstücke können in jede Programmiersprache eingebunden werden
- Festplattenunterstützung
- gutes Handbuch

#### Negativ:

- zur Zeit noch keine MIDI-Unterstützung





In der Record-Page kann man Musikstücke direkt über die Tastatur eingeben. Dabei sind auch Auftakte kein Problem

Editor werden einzelne Teile einer Melodie vorgegeben. Diese Mustervorgabe kann beispielsweise ständig wiederholt werden, wie man dies von Baß- und Schlagzeugbegleitung her kennt. Man muß also nicht immer wieder den gleichen Passus eingeben, wenn bestimmte Routinen verwendet werden. In den Patterns stehen die Noten und Statussymbole eines Musikstückes, die Makros – dazu gleich mehr – anwählen. Ein Pattern entspricht dabei einem Takt.

Wie bei den anderen zwei Editoren erleichtert die Cut-Copy-Paste-Funktion das Editieren ungemein. Diese Funktion sollte jedem Amiga-Besitzer ja vom Amiga-BASIC her bekannt sein. Was ist eigentlich ein Makro? Nun, ein Makro ist ein Programm, das ein Sample aufruft. In dem Makro-Editor werden also die Samples für ein Musikstück verwaltet. Der TFMX läßt bis zu 128 Makros, die in der Makro-Page editiert werden, zu. Grob gesagt ist die Makro-Page eigentlich nichts anderes als ein Editor für eine spezielle Programmiersprache, die Makroprogrammierung eben.

Der dritte Editor ist der sogenannte Track-Editor, in dem die einzelnen Patterns aufgerufen werden. Diese Patterns können an dieser Stelle transportiert werden. Beim TFMX stehen acht Tracks zur Verfügung, wobei natürlich eine kleine Einschränkung gemacht werden muß, da der Amiga ja nur vier Audio-Kanäle besitzt, also nur vier Töne gleichzeitig abspielen kann. Die weiteren vier Tracks erlauben jedoch eine flexible

und dadurch auch komplexere Gestaltung von Musikstücken.

**Da capo  
– aber al fine!**

Was kann der TFMX nun eigentlich alles? Nun, auf die verschiedenen Sample-Formate sind wir ja schon eingegangen. Die gängigsten Effekte wie Portamento, Fading, Vibrato, Echo etc. sind natürlich enthalten. Ferner unterstützt er die Multifunktionsfähigkeit des Amiga. Musikstücke können auch aus einzelnen Teilstücken zusammengefügt werden, was die Programmierung etwas leichter macht. Gleichzeitig sind bis zu 32 Songs verwaltbar. Als weiteres Positivum fällt auf, daß das Programm, das übrigens nur mit einem Hardware-Zusatz (Dongle) läuft, zwei Festplatten unterstützen kann.

Will man bereits erstellte Songs in eigene Programme einbauen, so ist dies auch kein Problem. Auf der Diskette ist ein Player vorhanden, der ermöglicht, daß Musikdaten auch eigenständig lauffähig sind. Das deutsche Handbuch ist ausführlich und gibt eine ausreichende Einführung anhand kleinerer Beispiele in den Umgang mit dem Editor. Ferner lassen sich Datendisketten vom Editor aus erstellen.

Leider ist das Programm noch nicht MIDI-fähig, es wurde jedoch angekündigt, daß das Update diese Option unterstützen soll. Ferner soll laut Hersteller ein Software-Sampler implementiert werden.

(br)



bytie  
präsentiert:

## Neue Dimensionen

### PERSONAL DISK

für IBM-kompatible PC/XT/AT

- 20 MB/40 MB Hard-Disk mit Tragetasche
- ideal zur Datensicherung und Datenaustausch
- anschliessbar ■ jedem PC mit Harddisk-Controller



### reis-mouse

Die Maus, die Ihren Computer schneller macht



für

- IBM™ PC/XT/AT und kompatible
- COMMODORE™ AMIGA,
- ATARI™ ST, SCHNEIDER™-EURO-PC/TOWER AT

■ Mechanischer Präzisions-Rollkugelantrieb  
Zwei optische Encoder, 4-Bit-CPU

### LESEN · STEuern · DRUCKEN

Die einzig wahre BARCODE-JEL-PAKET Lösung für Ihren PC:

Erkennung aller Codes z.B. EAN, UPC, JAN, Code 128 (ASCII !!), Interleaved 2/5, usw.

Generalvertrieb für Deutschland und Europa! Sofort Prospekt anfordern.

**Verkauf über den Fachhandel - Händleranfragen erwünscht**  
Info und Prospekte erhalten Sie direkt von






Postfach 38  
D-5584 Bullay  
Telefon 08542/2066-2067  
BTX • Reisware #  
Fax 08542/21017



Edgar Meyzies

# Überzeugendes Mathepaket

## MathTreasures-2.0

Seit Anfang des Jahres wird eine Mathe-Bibliothek in Modula-2 (M2AMIGA) angeboten, die eine wertvolle Ergänzung zu bereits veröffentlichten Modulen darstellt.

**W**as soll das? Nur eine Diskette, kein Handbuch und dafür über 100 DM berappen, das wirkt doch überzogen! Weit gefehlt! Die Diskette weist eine (gepackte) Dokumentation auf, die einen Drucker über 120 Seiten (einschließlich der Programmtexte der Schnittstellenmodule) ausspucken läßt. Was wird denn da eigentlich dokumentiert? Das Paket umfaßt 78 kurze Module, die die in Tabelle 1 dargestellten Bereiche abdecken. Die hohe Anzahl der Module besagt für sich alleine nicht viel, weder im Hinblick auf den Umfang der implementierten Funktionen, noch läßt sie Rückschlüsse auf die Handhabbarkeit zu. Bei näherer Betrachtung entdeckt man schnell, daß es sich eigentlich nur um wenige, thematisch zusammenhängende Modulpakete handelt, die jeweils auf die Arbeit mit einem Datentyp (zum Beispiel FFP, REAL, LONGREAL) ausgelegt sind. Die hohe Anzahl der Module ergibt sich also aus der Spezialisierung der Module auf Datentypen, aber auch aus dem Bemühen des Programmators um einheitliche Schnittstellen für die Funktionsprozeduren. Sie verrichten die gleichen Aufgaben, nur mit jeweils einem anderen Datentyp. So wird bei der Programmentwicklung der Wechsel zwischen Datentypen erleichtert, um zum Beispiel die Rechengenauigkeit zu erhöhen oder bewußt auf "speed" zu trimmen.

Die Implementierung der mathematischen Funktionen erfolgte sehr systematisch und konsequent. Allgemein hervorzuheben sind die anwenderfreundlichen Benutzerschnittstellen, die auf dem Amiga Bewährtes fortführen

und, wo immer möglich, in der Mathematik übliche Notationen zulassen. Das Paket ist durchaus für "ernsthafte" (frei nach dem Vornamen des Entwicklers E. Heinz) Anwendungen geeignet. Es schließt eine Lücke und macht dadurch Modula-2 auf dem Amiga noch attraktiver.

Zur Dokumentation könnte man auch die Texte der Implementationsmodule rechnen, die ausgedruckt nochmals etwa 200 Seiten füllen.

Ohne hohen Aufwand könnte die Beschreibung der Schnittstellen erheblich verbessert werden. Um professionellen Anforderungen zu genügen, sollten die Leistungsgrenzen der Module angegeben werden.

### Bruchrechnung:

Ein- und Ausgaben, alle arithmetischen Operationen, Konvertierung von Brüchen in Gleitpunktzahlen und umgekehrt

### Rechnen mit komplexen Zahlen:

Ein- und Ausgaben, alle üblichen Standardfunktionen für komplexe Zahlen der Form "Realteil + Imaginärteil \* i", implementiert für FFP, REAL und LONGREAL

### Umwandlung von Fließkommazahlen:

Typenübergänge zwischen FFP, REAL und LONGREAL

### Logische Funktionen:

Ergänzung der logischen Funktion NOT um die von BASIC her bekannten

### Ergänzungen/Erweiterungen der Math-Libs:

Komplettierung der MathLib0 durch eine neue MathLib1, die im wesentlichen zusätzliche Winkel- und Logarithmik-Funktionen bietet (getrennte Module für die Datentypen FFP, REAL und LONGREAL) sowie die Leistungsfähigkeit der jeweiligen Hardware voll ausschöpft.

### Arbeiten mit Matrizen:

Für die drei Datentypen (FFP, REAL, LONGREAL) jeweils 13 Module, um Matrizen beliebiger Größe anzulegen und damit zu rechnen. Dazu gehört auch das Lösen linearer Gleichungssysteme.

**Tabelle 1. MathTreasures-2.0 bietet für wichtige Bereiche der Mathematik eine reichhaltige und sichere Implementation leistungsfähiger Funktionen**

Die Quelltexte sind bestens geeignet, die effiziente Programmierung mathematischer Funktionen zu studieren. Die Module sind hervorragend strukturiert und sauber programmiert. Der Portierbarkeit scheint ein hoher Wert beigemessen zu sein. Die Übertragung der Matrix-Module auf einen anderen Rechner war innerhalb weniger Stunden abgeschlossen.

Herausstellen ist auch die hohe Zuverlässigkeit der Module und ihre Robustheit gegenüber fehlerhaften Eingaben. Die Programmlaufzeiten haben positiv überrascht. Coprozessoren werden optimal genutzt.

Schade, daß die Mathe-Bibliothek (noch?) nicht als "Library" im Sinne des Amiga implementiert ist und somit nicht von allen Sprachen aus nutzbar ist. Auch auf der Ebene der Objektdateien kann man sich nicht treffen, weil der Compiler des M2AMIGA (wie lange eigentlich noch?) sich die Exklusivität eines eigenen Linkformates leistet.

## Bruchrechnung

Bruch haben Sie gewiß nicht einzukalkulieren, wenn Sie die "MathTreasures" für diesbezügliche Rechnungen ein-

setzen. Sieben Modulpaare unterstützen diese häufig benötigte Form der Mathematik. (Haben Sie schon einmal an die Genauigkeit der Bruchrechnung gedacht?) Brüche werden als Zeichenketten eingelesen und so auch wieder ausgegeben. Alle üblichen Arten der Bruchrechnung (Addition, Subtraktion, Multiplikation, Division, GGT, KGV, Kürzen, Kehrwert usw.) werden unterstützt. Dabei kann mit unterschiedlicher Genauigkeit gerechnet werden, nämlich mit 16 und mit 32 Bit breiten (INTEGER und LONGINT) Bruchzahlen. Es wurde echte Integer-Arithmetik implementiert. Brüche können sowohl mit Bruchstrich als auch als Dezimalbruch (Dezimalzahl, als Bruch interpretiert) eingegeben werden. Sie erraten sicherlich schon die Struktur des Datentyps "Bruch"; er ist wie eine Zahl des Typs "REAL" mit einer Breite von 32 Bits implementiert, als varianter Verbund. An der Stelle des Dezimalpunktes kann auch der Bruchstrich vorhanden sein.

## Nur keine komplexen...

... Zahlen, bitte! Wer aber auf technisch-wissenschaftlichem Gebiet arbeiten muß, der findet in dem Paket "Complex" neun äußerst leistungsfähige Module für komplexe Zahlen der Form

"Realteil + Imaginärteil \* i".

Besondere Ein- und Ausgaberroutinen ermöglichen den Dialog mit dem Programm. Die Funktionen sind auf reelle Zahlen der Typen FFP, REAL und LONGREAL spezialisiert.

Neben den mathematischen Grundfunktionen (wie Addition, Multiplikation usw.) wurden weitere Funktionen (zum Beispiel Umwandlung in Polarkoordinaten und umgekehrt, Operationen komplexer Zahlen mit Gleitpunktzahlen, Kehrwertberechnungen usw.) implementiert. Es ist auch möglich, komplexe Zahlen aus Ausdrücken (reelle Zahlen) zu erzeugen. Alle mathematischen Ausdrücke wurden als echte Funktionsprozeduren implementiert, um eine Schachtelung in gewohnter Weise zu ermöglichen.



In der Modulvielfalt des Paketes geht "FloatConv" fast unter, obwohl es schon bei einfachen Anwendungen von geschwindigkeitsmäßig hohem Nutzen sein kann.

## Verflüssigen

Unter Ausschöpfung von Möglichkeiten der Library "Math-DoubTrans" wurden sechs Prozeduren zur Umwandlung von Fließpunktzahlen (FFP <-> REAL, FFP <-> LONGREAL und LONGREAL <-> REAL) zur Verfügung gestellt. Die Besitzer mathematischer Coprozessoren werden über die neuen Möglichkeiten besonders erfreut sein.

## Unkraut im Gerstenfeld?

Auch dem Modul "LogicLib" droht das Schicksal, übersehen zu werden. Das Modul stellt fünf Prozeduren (EQV, IMP, NAND, NOR und XOR) zur Verfügung, die der Compiler "m2c" nicht kennt. Wie mag das Modul nur in die Mathe-Bibliothek geraten sein, etwa wie eine Kornblume in ein Gerstenfeld? Nützlich ist es allemal!

Genau 15 Funktionsbibliotheken sind es, die die unter M2AMIGA implementierten Mathe-Module ergänzen bzw. sogar ablösen wollen. Die hohe Anzahl mag erschrecken, hatte

man es doch bisher mit wesentlich weniger Modulen zu tun.

## X-mal Math

Bei näherer Betrachtung erkennt man schnell, daß es eigentlich nur vier Gruppen von Modulen sind, die jeweils sehr ähnliche Aufrufkonventionen aufweisen:

- Ersatz der bereits bestehenden Standardmodule

- "MathLib0"
- "MathLibFFP" und
- "MathLibLong"

um die Sicherheit im Gebrauch zu erhöhen bzw. die Hardware (Mathe-Coprozessor) voll auszureizen. Die Anleitung enthält klare Empfehlungen zum Gebrauch. Die Library "Matheedoubtrans" wird sehr geschickt eingesetzt, um an Tempo zuzulegen.

○ Ergänzung der MathLib0 durch Implementierung zusätzlicher Standardfunktionen (zum Beispiel: deg, rad, rez, sqr, log dualis, Zehnerlog., Log. zu beliebiger Basis, pow, tan, cot, arcsin, arcos, arccot, sinh, cosh, tanh, coth, arcsinh, arcosh, artanh, arcoth) jeweils für die Datentypen FFP, REAL und LONGREAL.

○ Schneller Ersatz für die Standardbibliothek "MathLib0" durch Verzicht auf Überprü-

fung von Argumenten bei Aufruf der implementierten Funktionen und zur Ausschöpfung der Vorteile, die Mathe-Coprozessoren bieten. Die "Mathxxx-Module" zeichnen sich durch ihre Fehlerbehandlung aus. Aussagen zur Rechengenauigkeit fehlen jedoch.

## Im Viereck springen...

möchte man, wenn der "DIR-Befehl" 39 Namen von Modulen zur Arbeit mit Matrizen auf den Bildschirm knallt. (Ein Paukenschlag wäre angemessener, wie Sie gleich sehen werden.) Es sind jedoch eigentlich nur 313 Module, jeweils für die Datentypen FFP, REAL und LONGREAL implementiert. Nur wenige Programmiersprachen beherrschen die Matrizenrechnung. Für Modula-2 besteht nun eine äußerst effiziente Implementierung, die leicht auf andere Systeme zu portieren ist. Herausragend ist die dynamische Verwaltung der Matrizen, die beliebige Größen zuläßt und den Arbeitsspeicher geschickt nutzt. Das haben wir natürlich mit sehr großen Matrizen ausprobiert. No problem! Durch Verwendung dynamischer Datenstrukturen konnte eine besonders hohe Geschwindigkeit für die Ausführung der Matrixoperationen erzielt werden. Zuerst traut man seinen Augen nicht, wenn eine Matrix mit 100x100 Feldern ohne Mathe-Coprozessor für FFP-Zahlen in 1:52 Minuten invertiert wird. Für REAL-Zahlen ergaben sich 3:15 und für Zahlen des Typs LONGREAL 3:32 Minuten. Beeindruckend ist auch die mitgelieferte "3D-Demo", die mit Matrizenoperationen arbeitet.

Die Basismodule lassen eine Struktur erkennen, die auf Erweiterung ausgelegt ist. Alle Vorbereitungen sind getroffen, um beliebige Elementtypen zu implementieren. Wie wäre es zum Beispiel mit Vektoren oder Polynomen? Ein nächstes Update wird es erweisen.

Die Matrix-Module bieten die in Tabelle 2 dargestellte Funktionalität. Anschauliche Demoprogramme zeigen, wie unkompliziert es ist, die angebotenen Funktionen zu nutzen. Dazu eine kleine Kostprobe: Um eine Matrize anzulegen, sind nur zwei Prozeduren einzusetzen:

■ Mit dem Aufruf von "InitMatrix(testMatrix)" werden für "testMatrix" (vom Datentyp MATRIX) die Basisstrukturen angelegt, um die übrigen Funktionen darauf anzuwenden.

● Der Aufruf "DimAndFillMatrix(testMatrix,z,s,0.0)" bewirkt, daß unser Testobjekt mit "z" Zeilen und "s" Spalten angelegt sowie jedes Element mit "0.0" initialisiert wird.

Auf die "LGSLib" (Lineare Gleichungs-Systeme) möchten wir Ihr besonderes Interesse richten. Das Modul stellt sehr mächtige Routinen zum Lösen praktisch beliebig großer linearer Gleichungssysteme zur Verfügung. Die Routinen bewährten sich in volkswirtschaftlichen Simulationsmodellen. Sie lieferten stets korrekte Ergebnisse und überzeugten durch ihre hohe Arbeitsgeschwindigkeit.

## Schätzchen oder Schatz?

Nach mehrtägiger Arbeit mit vielen der angebotenen Funktionen fällt die Bewertung der "MathTreasures" mit "echter Schatz" sehr leicht. Faszinierend ist die Leistungsfähigkeit der "Complex-" und der "Matrix-Module". Bereits als Schüler und als Student hätte ich gerne ein solches Paket besessen. Heute unterstützt es mich im Bereich der Simulation auf ausgezeichnete Weise.

(mm)

Modul	Enthaltene Funktionen
"MatrixArith":	arithmetische Standardoperationen
"MatrixBase":	Matrixerzeugung und -verwaltung
"MatrixCopy":	Kopieren von Matrizen und Matrixteilen
"MatrixDecs":	zentrales Deklarationsmodul
"MatrixDiskIO":	Laden und Speichern von Matrizen auf Diskette
"MatrixElemOps":	Operationen auf allen Matrixelementen
"MatrixIO":	I/O für Matrizen auf Grundlage von "InOut"
"MatrixMinMax":	Minimum-/Maximumbestimmung
"MatrixNorm":	Normberechnung für Matrizen
"MatrixOps0":	diverse Zeilen- und Spaltenoperationen
"MatrixOps1":	Gauß-Algorithmus, Transponierung, Determinanten und Rangbestimmung
"MatrixOps2":	LR-Zerlegung und Matrixinvertierung
"LGSLib":	Lösen beliebiger linearer Gleichungssysteme

Tabelle 2. Für jeden der drei Datentypen FFP, REAL und LONGREAL bietet "MathTreasures-2.0" mächtige Funktionen ■■■ Bearbeitung von Matrizen

## AMIGA DOS Blitzlicht

**Name:** MathTreasures-2.0  
**Hersteller:** A + L AG, Schweiz  
**Vertrieb:** Fachhandel  
**Preis:** 100,- DM

### Positiv:

- Installation auf Festplatte möglich
- Einbinden in eigene Programme
- hohe Genauigkeit
- sehr schnell
- ausführliche Dokumentation
- einheitliche Schnittstellen
- gutes Preis-Leistungsverhältnis

### Negativ:

- noch nicht als Library implementiert
- eigener Linker



# Sendung läuft!

## Von Computer zu Computer – Kermit hilft

Bevor Sie jetzt die Nase rümpfen, weil Sie glauben, wir wollten Ihnen etwas über die Sesamstraße erzählen, ist es besser, Sie über Kermit aufzuklären. Kermit ist ein Public-Domain-Programm, das unter Zuhilfenahme eines seriellen Nullmodem-Kabels Daten über die serielle Schnittstelle überträgt. Wie Sie verstehen nur Bahnhofshalle, Ausgang Nord? Na, dann lesen Sie doch mal weiter, das Ganze hat nämlich einen Sinn.

AMIGA DOS

Bei einem Amiga-Besitzer kann es aber auch sein, daß sie beruflich und privat mit den "Konkurrenten" des Amiga zu tun haben, nämlich ST und PC. Bei beiden gibt es einen großen Hindernisgrund, Daten und Dateien von den jeweils anderen Disketten zu kopieren – das Nullmodemformat. Und jetzt kommt Kermit ins Spiel – natürlich nicht der Frosch, sondern das Public-Domain-Programm. Und die serielle Schnittstelle unseres Amiga, die bisher den Architekturstatus besaß, bekommt plötzlich einen ganz neuen Stellenwert.

### Noch ein Bit und noch ein Bit – seriell ist nicht schnell

Um Kermit sinnvoll nutzen zu können, brauchen wir noch ein klitzekleines bißchen Hardware – das Nullmodem-Kabel.

Bei den meisten Lesern wird gerade jetzt das Argument "Modem" irgendwo im Hinterkopf irgendeine Assoziation hervorrufen, bei der Konstellation "Nullmodem" wird jedoch so mancher erst einmal erstaunt aufstöhnen. Das Wortende "Kabel" erzeugt dagegen bei "Nichtbastlern" Schauerwellen. Dabei ist das alles gar nicht so wild, wie es sich bis jetzt angehört hat.

Bei einer seriellen Datenübertragung wird jedes Byte, das vom Sender zum Empfänger geschickt wird, durch eine bestimmte Anzahl

von Parameter-Bits umhüllt. Damit läßt sich erreichen, daß der Empfänger den Sender "mitnimmt". O.K., alles richtig angekommen! oder "Das war wohl nix! Nochmal!!". Ein solcher Vergleich der abgesendeten Daten mit den empfangenen heißt in der Fachsprache Übertragungsprotokoll.

Heutzutage wird viel über DFÜ gesprochen (Datenfernübertragung), und diese DFÜ wird in der banalsten Art benutzt. Bekannteste Möglichkeiten der DFÜ sind Datenübertragung über Modems (MODulator – DEModulator) oder Akustikkoppler. Im Grunde genommen findet hier ein Kreislauf statt, wobei Sender und Empfänger ihren Status wechseln, der Sender also auch als Empfänger und der Empfänger auch als Sender benutzt werden kann. Es geht auch anders, man kann Sender und Empfänger bestimmen, dies soll uns jedoch hier nicht interessieren.

Bei der Sendung findet also eine Umwandlung unserer Daten in eine Artbits statt, damit diese korrekt zum Empfänger gemeldet werden können. Der wiederum wandelt die empfangenen Daten in normale Bytes um – wenn nicht bei der Übertragung etwas Ausfallerregendes ist. Wie schon gesagt, wir können uns das Ganze als einen Datenkreislauf vorstellen, in dem zwei Computer über ihre serielle Schnittstelle und Umwandlungsgeräte miteinander kommunizieren.

Warum aber seriell und nicht parallel? Modems und Ak-

ustikkoppler nutzen eine Verbindung aus, die wir ebenfalls kennen, das Fernmelde-netz. Dieses besteht im Grundeprinzip nur aus zwei Leitern, der Datenleitung und der Masseleitung. Würden wir parallel übertragen, wären schon acht Datenleitungen erforderlich, um alle Zeichen nutzen zu können. Erinnern wir uns: Jedes Byte hat eine Länge von acht Bit, jeder Zeichencode läßt sich in einem Byte unterbringen. Damit haben wir die Möglichkeit, 255 Zeichen in einem Byte (218 = 256 - 1) unterzubringen. Da jedes Bit einen Stellenwert hat (210, 211, 212, usw.), brauchen wir für jeden Stellenwert auch eine Leitung. Das geht aber nicht, denn, wie gesagt, wir haben nur zwei. Splitten wir die Bytes in Bits auf und schicken sie hintereinander los, kann dann wir tatsächlich nur zwei Drähte (bzw. acht) und ein Gerät, was aus den zerstückelten Bytes wieder ganze macht.

Da gibt es nur eine Lösungsmöglichkeit. Der "Zusammensetzer" muß wissen, ob das, was er zusammensetzt, auch so richtig ist. Er muß sich also beim Sender erkundigen, ob das zusammengesetzte Byte genauso aussieht wie das zerstückelte. Und jetzt kommt das Übertragungsprotokoll zur Geltung. Es bringt quasi als Bote Informationen über die abgesendeten und empfangenen Bits vom Empfänger zum Sender und sorgt damit für einen fehlerfreien Ablauf.

Der Ablauf bei der seriellen Datenübertragung sieht also vereinfacht so aus:

Unter den Amiga-Besitzern gibt es viele, die schon vorher einen Computer das Programmieren beizubringen hatten, viele 8-Bit-Geräte hatten seit dem Einbruch des Amiga als Requisite oder Museumsware ungenutzt in der Ecke. Kein Problem, wenn da nicht die Datei über die aufgenommene Videofilme oder die Dis-Sammlung auf einer zerstückelten Diskette existieren würde, die niemand Amiga absolut nicht lesen will. Was tun? Den kleinen Vorgänger an die andächtige Commodore Generation verschenken (unabhängige Klammern sind geradezu glücklich, wenn sie ein Es-High-Tech-Garab aus-einandernehmen dürfen), die entsprechenden Disketten zu diversen Weihenachtgeschenken umtauschen (Disketten-schönheiten, Disketten-Spandosen, Disketten-Schokolade)? Eigentlich schade, wenn man bedenkt, wieviel Arbeit und Zeit damals investiert wurden, um sich mit dem Thema Computer vertraut zu machen.



Sender -> Daten -> Umwandler (Modem) -> Kabel -> Umwandler

Empfänger -> Protokoll -> Sender -> entweder o.k. oder neuer Versuch -> Empfänger

Die jetzt haben wir einen gravierenden Nachteil der seriellen Datenübertragung mit dabei. Sie ist zu langsam. Einzelstellige Zeichen wir aber auch ein ganzes Verschiebegerät. Fernmeldetzute gibt es überall auf der Welt. Einen Datenbestand von Hamburg nach New York zu schicken, kostet höchstens ein müdes Lächeln (und

bestens Daten ein Handen Info-Schrott wird, die bestenfalls noch im Trashcan ein Plätzchen finden.

Bleibt also nur, den Sender und Empfängerinnen getrennte Leitungen zu spezialisieren, macht mit UNB (Unidirectional Bus) auch Grund oder Masse ankommt drei Verbindungen. Das war aber nicht nicht alles. Ist unser Amiga Kompatibel, muß er beim anderen Rechnern, ob und wann Daten kommen, dies geht ebenfalls nicht über die Datenleitung (oder Drähte). Ist der Amiga Sender, muß er dem Empfänger mitteilen, daß jetzt 'Quinn auf' gilt, dies geht je

IGA DOS

Teilungsbildern bitte weiter liefern.

Die Datenübertragung mit wenigen Worten zu beschreiben ist nicht möglich, dazu ist das Thema zu umfangreich. Für unsere Zwecke jedoch soll es erst einmal stehen, denn wir brauchen nicht allzuviel Verständnis der Hardware beim Umgang mit Kermit.

## Was überträgt Daten ohne Modem? Ein Nullmodem-Kabel

Mit den empfindlichen Geräten haben wir quasi einen kompletten Datenkreislauf errichtet (auch wenn nicht unsere Daten zurückkommen, sondern anders). Aus diesem Kreislauf können wir jetzt zwei Teile isolieren: das Modem und das Übertragungsmedium, also das Fernmeldenetz. übrig bleiben nur die Anschlußkabel am Computer. Diese zwei können wir jetzt miteinander verbinden und schon...

Nullmodem-Kurzschluß - oder besser gesagt, Nullmodem-Kabel. Mit der Zweidrahttheorie ist das nämlich eine Sache. Nach dem Modem wäre es ja kein großes Problem (bzw. es ist auch da mehr als zwei Drähte, aber das ist Inhalt eines anderen Artikels). Vor dem Modem jedoch müssen wir unterscheiden: Wird der Daten getrennt worden, oder werden sie kombiniert? Beides möglich ist, einem Drähtpaar eine Modemfunktion zu dem Grunde, daß aus unseren

noch wiederum nicht auf die 'Empfängerseite' (bzw. wenn Sender und Empfänger auf gleichem Anschluß gegenüber den Ansichten zu sein, kann ebenfalls nichts funktionieren (bzw. Drähte). Nun, um die ganze Sache abzukürzen: Ein Nullmodem-Kabel hat mehrere Verbindungen, um die Daten mit anderen Computern auszutauschen. Um zwei Computer direkt miteinander zu verbinden, ist es so, die beiden einen der einen mit der Anschlußleitung des anderen und die Sendeleitung des einen mit der Empfangsleitung des anderen zu verbinden. Die Kontrolleleitungen müssen jetzt noch so geschaltet werden, daß beide Rechner den jeweiligen Zustand selbst erkennen. Ein solches Kabel nennt man Nullmodem-Kabel, eben weil es ohne Umwandlung durch das Modem auskommt.

Die Pinbelegung des entsprechenden Kabels finden Sie in Abbildung 1. Dieses Kabel sollte eigentlich mit fast allen Computern funktionieren, die über eine RS232C (V.24) Schnittstelle verfügen. Bevor Sie jetzt losgehen, Sie finden sie unter der Bezeichnung 'Kermit' auf der Rückseite Ihres Amiga 500, 1000 oder 2000. Beim Amiga 1000 gibt es noch dazu noch ein Problem. Seine Schnittstelle entspricht in der Pinbelegung nicht der Norm (sonst man von einer Standard-Nullmodem-Kabel sprechen kann, die Signalleitungen nämlich gegeneinander an. Für den Anschluß des Nullmodem-Kabels ist aber auch noch die intensive Suche nach dem

richtigen Anschlußbild nötig. Sie haben übrigens richtig gelesen bei 'sollte funktionieren'. Leider meinen manche Computerhersteller mit 'Null' ihre eigene, und so kann es schnell passieren, daß die 'Kompatibilität' (bzw. 'Kompatibilität') nur ausreicht, kompatibel ist. Mit unserem Kabel hat es jedoch einwandfrei zwischen einem Amstrad PC 1512 und einem A2500, sowie einem Peacock AT und einem A500 funktioniert, so daß es fast auszuschließen ist, daß hier Probleme entstehen. Falls nicht, gilt's nur die 'Wortwahl', zu experimentieren ist eine denkbare Hilfe wäre zum Beispiel das Entfernen der Leitungsumschaltungen 4 -> 5 und 3 -> 4, wobei die Pins 5 und 4 im Stecker vertauscht werden müssen. Soweit zum Lötteil, jetzt kommt endlich die Software.

## Das Schönste, was man mit seriellen Schnittstellen machen kann, ist übertragen (O-Ton Kermit)

'Kermit' ist eine Wortzusammensetzung aus der keltischen Sprache und bedeutet 'umsonst' (bzw. 'umsonst' 'Public Domain?'). Ähnlichkeiten mit einem berühmten Fernsehstar sind also nicht festzustellen. Festzustellen

ist aber eines: Kermit ist für sehr viele Computer vorhanden, darunter sogar für VAX- und UNIX-Arten im Großrechnerbereich. Interessant wären für unsere Rechner die CP/M-2.0- oder CP/M-3.0-Versionen. CP/M ist ein Betriebssystem, was in diesen Versionen auf vielen 8-Bit-Computern zu finden ist, dazu gehören zum Beispiel die Amstrad CPC und Joyce, der Commodore 128, in der CP/M86-Version auch für viele 16-Bit-PCs. Atari ST und MS-DOS scheinen ebenfalls auch zum Kermit-Übersetzer 'MSKERMIT' (bzw. schon in Versionen bis 3.0 vor statt nicht noch höher). Damit ist ein Großteil der Amiga-Vorgänger und -Mitbewerber schon abgedeckt.

Sicher, für den Amiga gibt es inzwischen schon eine Menge neuer (und vielleicht auch besserer) Übertragungsprogramme, jedoch hat keine die Verbreitung erreicht wie Kermit. Ein Grund mehr also, sich das Programm anzusehen. Da es außerdem Public Domain ist, besteht kein Grund, einen Kauf nicht zu tätigen.

Durch die weite Verbreitung von Kermit läßt sich auf verschiedenen Computern trotz einer Angleichung vornehmen, was bei unterschiedlichen Computertypen sonst kaum möglich ist.

Was kann nun Kermit? Kermit läßt sich von der Workbench wie auch von der Shell



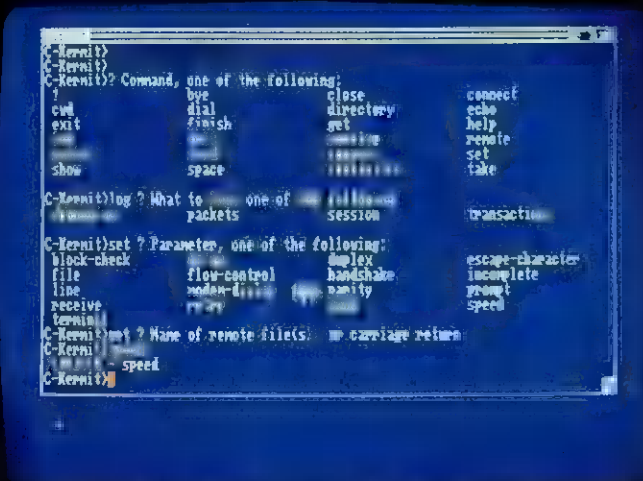


Abbildung 1: Die Befehle von Kermit. Ob der Rechner nun als Server, Sender oder Empfänger fungiert, sie haben alle Einstellungsmöglichkeiten.

beim ersten Einlesen. Dabei ist eine Vielzahl an Einstellungsmöglichkeiten. So läßt sich die Baudrate einstellen, aber werden alle Werte akzeptiert. Sinnvoll ist es jedoch, erstens nur Standardwerte einzugeben und zweitens beide Computer, Sender und Empfänger, auf die gleiche Baudrate anzupassen (Baud = Bit pro Sekunde). Kermit kann im Voll-Duplex-Modus arbeiten,

das heißt, Daten können gleichzeitig empfangen und gesendet werden. Infall existieren auch getrennte Sende- und Empfangskanäle, ein Zusammenstoß ist damit ausgeschlossen.

Überhaupt ist Kermit sehr anwenderfreundlich. Durch die Nutzung der <SHIFT> + <?> Tasten werden alle Kommandos aufgerufen. Gibt man das Kommando mit Fragezeichen

## Liste der Bauteile

2 SUB-D-Buchsen 25poliger Lötanschluß  
2 Gehäuse für 25polige SUB-D  
8poliges abgeschirmtes Kabel  
in Rundausführung (bis zu 5 Meter)

Als Buchsengehäuse sind 'Posthauben' ideal (Metallausführung), die Abschirmung sollte elektrisch mit dem Metall verbunden werden.

ein werden die erwarteten Parameter angezeigt. Fehler werden sofort erkannt, führen also nicht zu Problemen. Allerdings, um nicht dem Standard entsprechen, werden zwar akzeptiert, die Benutzer wird jedoch darauf hingewiesen, daß die eben eingegebene Baudrate ungewöhnlich ist.

Kermit läßt das Einlesen des jeweiligen Datenmaterials zu, es wird dann wie im DOS benutzt. Die zu übertragenden Dateien sollten im ASCII-Code vorliegen. Steuerzeichen werden beim Protokoll erkannt und interpretiert, aber trotzdem ignoriert, was auf dem Empfänger recht eigenartige Textpassagen erzeugen kann. Probleme beim Übertragen lassen sich jederzeit abrufen, unbekannte Phänomene fallen damit in die Kategorie 'versteht'.

Interessant ist auch die Möglichkeit, Computern als 'Server' einzusetzen. Dazu ein einfaches Beispiel:

Durch GET BAUD 19200 auf beiden Rechnern die Baudrate setzen, dann CONNECT eingeben, ebenfalls auf beiden Computern – und schon kann man auf Tastatur 1 etwas eingeben, was gleichzeitig auf Monitor 2 zu sehen ist und umgekehrt.

Wie gesagt, Kermit enthält viele Funktionen, alle aufzuzählen, ist ein schwieriges Unterfangen. Nur soviel: Wer sich ein Nullmodem-Kabel baut (zu kaufen sind sie nicht, mit zirka 20 bis 30 DM etwas teurer) und sich die notwendigen Verbindungen für einen seriellen Computer zugeht, hat gute Chancen, seine alten Daten zu retten oder Arbeit von fremden Computern auf dem Arbeitsplatz zu besorgen, vorausgesetzt, der 'Ex' verfügt ebenfalls über eine RS-232C-Schnittstelle. Sicher, die A2000- (oder höher) Rechner haben ihren PC-Teil mit AREAD und AWRITE, und somit bietet ein Übertragungsprogramm mit Kabel mehr Komfort. Diejenigen, die einen A400 oder A1000 besitzen, sind da auch besser bedient.

Ach übrigens – manche Spiele lassen einen Zweifachspieler-Modus über die serielle Schnittstelle zu; Was müßte Sie, wie immer, es sich leisten, mit seinem Mitstreiter über zwei Rechner zu kommunizieren. Auch der Flight Simulator II bietet diese Möglichkeit an. Mal sehen, von wem das Ziel kommt. Überhaupt bekommen selbst Flugsimulationen wie T13 Strike Eagle andere Dimensionen, wenn der Computer plötzlich nicht mehr der Computer, sondern jemand aus der Familie ist und Sie geradeheraus per Kabel durch das imaginäre Luftloch gegenseitig eingelegte Passwörter kennen da nicht bekommen. Ihren engsten Verwandten eine Kabine nachtschicken.

(16)

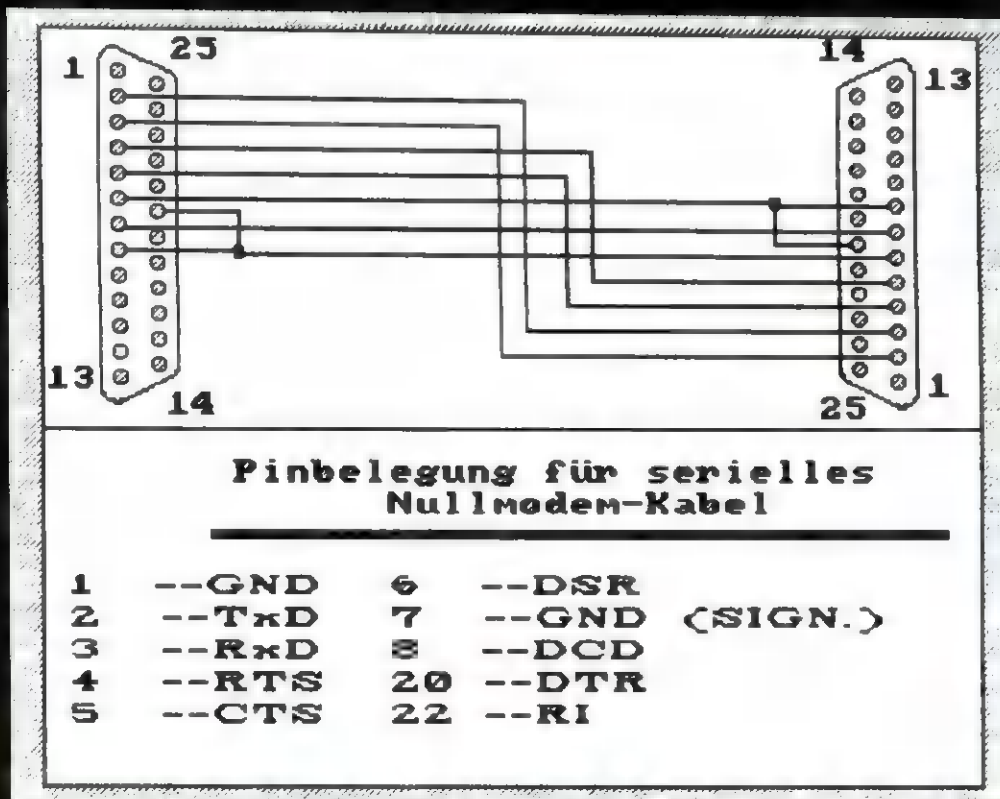


Abbildung 2: Die Zeichnung des seriellen Schnittstellenkabels. Benötigt werden nur wenige Bauteile.



Markus Steimar

# Basteleien, nicht nur für Anfänger

## Reset und Prozessorbremse für Amiga 500 und 1000 im Eigenbau

**Hardwarebasteleien können Abenteuer werden. Jemand, der noch nie in seinem Leben einen Lötkolben in der Hand hatte, verspürt angesichts im Eigenbau aufgemotzter Hardware einen Hauch von Ehrfurcht gegenüber dem versierten Hardwarebastler. Allein wie richtig gelötet wird, wird für den Laien zum Problem. Was wirklich fehlt, ist ein vernünftiges Projekt und eine genaue Anleitung...**

**W**o fang ich an, fragt sich der Interessierte, überlegt und besorgt sich zunächst einmal einen Lötkolben. Nur welches Fabrikat? Welche Leistung? Das Angebot ist vielfältig.

Tatsächlich gibt es für unterschiedliche Aufgaben verschiedene Lötkolben. Der Anfänger sollte sich einen Kolben mit 25 Watt Leistung und einer Feinspitze zulegen. Alle Geräte mit höherer Leistung eignen nur zum Weißblechschneiden und noch rabiatere Anwendungen. Zu unserer Bastel-Grundausstattung gehören allerdings noch ein paar weitere Kleinigkeiten: Lötzinn und eine Ablötpum-

pe. Also besorgen wir uns Feinstlötzinn mit mindestens 60% Zinnanteil. Ergänzt wird das Ganze durch eine gute Schreibtischlampe (wir wollen sehen, was wir löten), eine saubere Arbeitsfläche und einen bequemen Stuhl. Nun ist die benötigte Ausrüstung beisammen; was uns fehlt, ist ein Projekt, das nicht zu umfangreich ist, aber trotzdem sinnvoll sein muß. (Ein wesentliches Motivationsmoment, oder geben Sie sich beim Produzieren von Unsinn Mühe?)

Als sinnvolle kleine Hardwareprojekte würden sich ein Resetschalter und eine Ablaufbremse für unseren Amiga nicht schlecht machen. Tatsächlich ist beides recht einfach zu realisieren, es brauchen am Erweiterungsport des Amiga nur ein paar Drähte mit bestimmten Pins verlötet zu werden. Die Drähte werden mit einem Schalter verbunden, der bei Betätigen die entsprechende Funktion auslöst. Einzelheiten über die

benötigten Bauteile entnehmen Sie bitte der Stückliste.

Nächster Schritt im Zuge unserer Hardwarebasterei ist das Studium des Bauplans. Hier gilt es, sich zunächst einmal zurechtzufinden. Der Amiga-Erweiterungsport besteht aus 85 Kontakten. Wo Kontakt eins und wo Kontakt 85 liegen, können Sie Ihrer Amiga-Dokumentation entnehmen (bitte achten Sie darauf, daß die Expansionsports des Amiga 500 und die des Amiga 1000 umgekehrt sind).

Wir könnten nun mit der Arbeit beginnen und unsere Drähte an die entsprechenden Kontakte löten. Nur werden Sie irgendwann einmal solch voreiliges Handeln verfluchen. Zum Beispiel dann, wenn Sie eine Speichererweiterung oder einen Festplattencontroller an den mit Drähten und Lötzinn verschönten Expansionsport anschließen wollen. Und damit haben wir auch schon eine der elementaren Regeln entdeckt, die bei Hardwarebasteleien dringendst beherzigt werden sollten. Zunächst sollte das ganze Projekt genau überdacht werden, insbesondere mögliche Schäden sollten vor Arbeitsbeginn ins Kalkül gezogen werden. Voreilige und Unbedachte haben schnell, sehr zur Freude der Hersteller, die teure Hardware ruiniert und müssen kostspielige Reparaturen in Kauf nehmen. Uns hilft ein DMA-Portexpander weiter. Diese Erweiterung wirkt im Grunde wie eine Weiche für den Port. An dem Portexpander befinden sich fünf, jeweils einzeln anschaltbare Ports. Wir können also unsere Lötarbeiten an einem dieser Ports vornehmen, ohne den Amiga zu gefährden. Ist alles fertig, muß der Portexpander nur noch an den Expansions-

port des Amiga angeschlossen werden.

Doch zurück zu unserer eigentlichen Basterei. Wie Sie auf den Bauplänen sehen, werden bei der Bremsenschaltung am Amiga 500 und am Amiga 1000 die Pins 1 und 3, beides Masseleitungen, miteinander verbunden. An Pin 3 wird nun ein Draht angelötet, der mit einem Kontakt des Schalters verbunden wird. Ebenso wird an Pin 55, der Halteleitung des Prozessors, ein weiterer Draht angelötet, dessen anderes Ende mit dem freien Kontakt des Schalters verbunden wird. Auch bei der Resetschaltung werden bei Amiga 500 und 1000 die Masseleitungen von Pin 1 und 3 miteinander verbunden. Wie schon bei der Bremse wird nun an Pin 3 ein Draht angelötet und dessen anderes Ende mit einem Kontakt des Tasters verbunden. Ein zweiter Draht wird an Pin 53 angelötet und mit dem anderen Kontakt des Tasters verbunden. Es ist durchaus möglich, beide Basteleien an einer Porterweiterung zu installieren. Es sollte nur dringend darauf geachtet werden, daß zwischen den Drähten, die an Pin 53 und 55 angeschlossen sind, kein Kontakt zustandekommen kann. Was die Pins 1 und 3 betrifft, ist es eigentlich egal, an welchem von beiden der Draht angeschlossen wird, da es sich bei beiden um Masseleitungen handelt.

(hs)

### Stückliste für Resetschalter und Bremse am Amiga-Expansionsport

- 1 Stück Amiga-DMA-Portexpander,
- 1 Stück Kipphebel- oder Wippenschalter,
- 1 Stück Taster

Bitte beachten Sie, daß bei unsachgemäßer Handhabung Ihres Computers ein eventuell noch bestehender Garantieanspruch verfallen kann. Für Schäden, die durch eventuelle Fehler in den abgedruckten Bauanleitungen entstehen, kann weder die Haftung noch eine juristische Verantwortung übernommen werden.

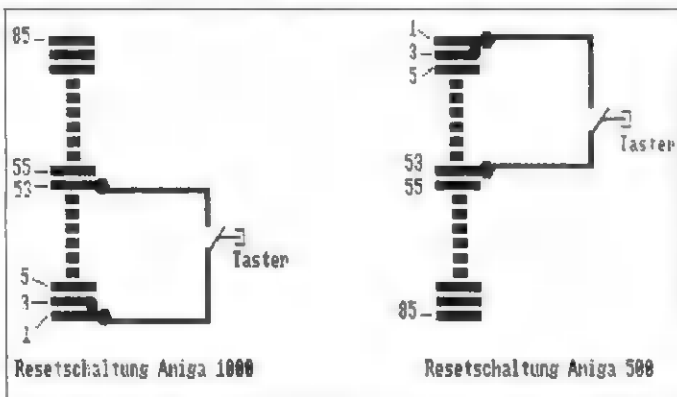


Bild 1. Bauplan für die Resetschaltung am Amiga 500 und 1000

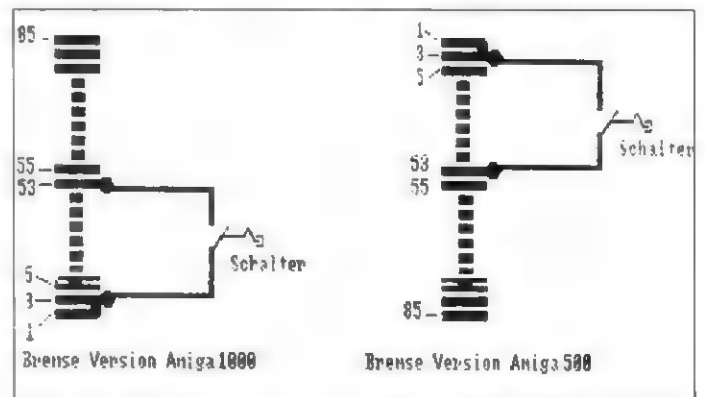


Bild 2. Bauplan für die Bremsenschaltung am Amiga 500 und 1000





Drucker mit einer seriellen Schnittstelle zu versehen. Besondere Aufmerksamkeit erregt ein kleines Sensor-Tastenfeld mit LCD-Anzeige auf dem Drucker. Der Papiereinzug kann entweder von hinten oder aber von unten vollzogen werden. Dies ist besonders wichtig, wenn man Platzprobleme hat. Wie sieht nun das Innenleben des Swift 24 aus?

### Außen hui und innen...?

Mit einem einzigen Handgriff läßt sich der Swift 24 aufklappen und zeigt sein Inneres. Ohne größere Probleme läßt sich das Farbband einsetzen. Der Druckkopf läßt sich ebenso leicht und schnell wechseln. Der Einzelblatteinzug erfolgt wahlweise manuell oder automatisch. Die Einzelblattführung läßt sich leicht aufsetzen, so daß man die einzelnen Blätter dann problemlos in den Schacht einführen kann. Auffällig beim Innenleben des Swift 24 ist ein kleines Fach an der rechten Seite des Druckers, wo sich ein Steckplatz für Schriftartkarten befindet, der dem Drucker eine größere Flexibilität und Vielseitigkeit verleiht.

Ein weiterer Punkt, der positiv zu bewerten ist, ist die Möglichkeit, Farboptionen zu wählen, um so in sieben verschiedenen Farben zu drucken. Man benötigt dazu ein mehrfarbiges Farbband und eine maschinelle Farbeinheit. Der temporäre Speicher des Druckers läßt sich mit einem RAM-Chip von den standardmäßigen 8 kByte auf 40 kByte erweitern. Diese Zusatzoption lohnt sich in erster

## Auf die Plätze, fertig, Druck

### Citizen Swift 24 – ein Sprinter unter den Druckern

Beim Kauf eines Druckers hat der Käufer eines immer im Auge – "Value for money". Das heißt, er möchte für sein "sauer Ersparnis" natürlich auch den entsprechenden Gegenwert haben. Die Firma Citizen hat mit dem Swift 24 einen 24-Nadel-Drucker auf den Markt gebracht, bei dem man wirklich "Value for money" bekommt.

**N**ehmen wir den Citizen Swift 24 etwas genauer unter die Lupe. Von den Ausmaßen her gehört dieser Drucker zu den kleineren. Auch sein äußeres Erscheinungsbild läßt sich am treffendsten mit handlich, kompakt beschreiben. Die Buchse für den Netzstecker

liegt an der rückwärtigen Seite. Der parallele Anschluß führt über eine Buchse, die an der rechten Seite des Druckers angebracht ist. Oberhalb dieser Buchse besteht die Möglichkeit, den



Bild 1. So sieht das Originalbild "Cloudnine" auf dem Amiga aus

Abbildung 1. Der Negativ-Ausdruck des 4096-Farben-Bildes "Cloudnine"



Linie dann, wenn größere Dokumentmengen verarbeitet werden sollen. Gehen wir noch etwas mehr auf die Leistungsfähigkeit dieses Druckers ein.

## ... ein leistungsfähiger Sprinter

Wie läuft die Konfiguration des Swift 24 ab? Hier tritt das Sensortastensfeld mit seiner LCD-Anzeige auf den Plan. Sechs verschiedene Tastenfelder machen die Konfiguration auch ohne Handbuch durch die gleichzeitige LCD-Anzeige fast zum Kinderspiel. So hat man immer den Überblick darüber, was gerade eingestellt wird und in welchem Untermenü man sich befindet. Weiterhin bestehen verschiedene Möglichkeiten zur Konfiguration: Zum einen kann von Menü zu Menü jeder Punkt einzeln angesprochen und ausgewählt werden. Schneller geht's jedoch mit dem Quick-Set-up, wo Makros angesprochen werden können; der Drucker bleibt dabei auf online geschaltet. Vier verschiedene Makros können dabei definiert werden, um die jeweilige Konfiguration darin zu speichern.

Eine Tastenfunktion fällt besonders ins Auge: die PARK-LOAD-Funktion. Sie dient dazu, die Seitenanfangsposition und den Mikro-Zeilenvorschub zu definieren und zu speichern. Ferner kann man mit dieser Funktion die Abreißposition sowie die horizontale Punktaussteuerung ganz genau bestimmen und natürlich auch speichern. Kommen wir zu einem weiteren sehr positiven Feature. Die Tatsache, daß Geschwindig-

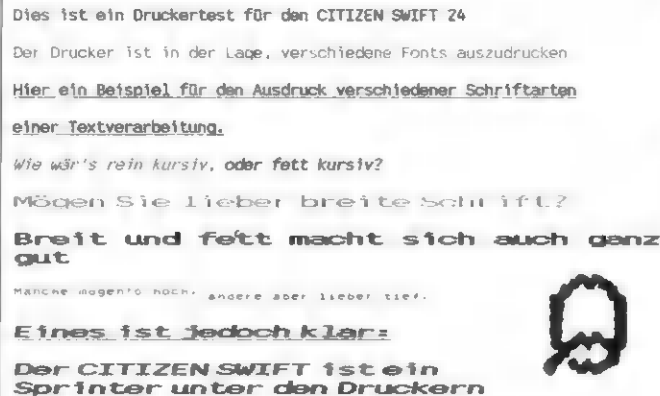


Abbildung 2. Eine Druckprobe des Swift 24, bei der verschiedene Schriftarten einer Textverarbeitung dargestellt werden

keit keine Hexerei ist, bewahrt sich beim Swift 24 auf sehr eindrucksvolle Weise. Der Probedruck der verschiedenen Schriftarten (Abbildung 1) erfolgte in "Null Komma nichts". Auch die Geschwindigkeit, mit der der Negativdruck des 4096-Farben-Bildes "Cloudnine" (Abbildung 2) im LQ-Roman-Modus abgedruckt wurde, war ein überzeugender Beweis seiner Leistungsfähigkeit. Als Vergleich kann das Originalbild herangezogen werden (Bild 1). Der schnellste Modus ist der Draftmodus, auch Entwurfsmodus genannt. Hier werden die Zeichen mit einer Geschwindigkeit von 192 Zeichen pro Sekunde aufs Papier gebracht.

Insgesamt vier verschiedene Schriftarten stehen dem Benutzer zur Verfügung: Times Roman, Helvetica, Courier und Prestige. Mit weiteren Features, wie den Sonderschriftarten Outline, Shadow und Reverse, lassen sich Texte

grafisch etwas aufpeppen. Die bereits erwähnten IC-Schriftarten sind optional und gehören somit nicht zum Lieferumfang. Auffällig ist, daß der Citizen Swift 24 über unterschiedliche Emulationen verfügt. Der Swift kann dabei den Epson LQ 850, den IBM Proprinter und den NEC P6 emulieren. Sie können entweder per Tastensfeld eingestellt oder mit Hilfe von Escape-Sequenzen vom Computer an den Drucker geschickt werden.

## Gesamteindruck positiv

Die Startposition des Druckkopfes wird mit einem Mikroschalter abgetastet und kann der Präzision des Druckwerks nichts anhaben. Beim Drucken selbst ist die Geräuschkulisse sehr hoch, da das Gehäuse und das Druckwerk nicht schallgedämpft sind.

Im Bereich der 24-Nadler fällt der Citizen Swift 24 schon et-

was aus dem Rahmen. Für einen Preis von 1098,- DM liefert er eine ganze Menge Quantität und auch Qualität, wo bei anderen Druckern dieser Preisklasse doch hier und da noch einiges im Argen liegt. Das einzige Manko dieses Druckers ist zugleich ein wenig unangenehm: Die Geräuschkulisse ist relativ hoch. Hier könnten sich die Hersteller vielleicht noch etwas einfallen lassen. Ansonsten wird der sehr positive Eindruck, den der Swift 24 hinterläßt, in keinsten Weise geschmälert.

(vb)

## AMIGA DOS Blitzlicht

**Name:** Citizen Swift 24  
**Art:** 24-Nadel-Matrixdrucker  
**Hersteller:** Citizen Europe Deutschland  
8056 Neufahrn  
**Vertrieb:** Fachhandel  
**Preis:** 1098,- DM

### Positiv:

- sehr schnell
- verfügt über drei unterschiedliche Emulationen
- PARK-LOAD-Funktion
- Sonderschriftarten Outline, Shadow, Reverse
- leichte Konfigurierung durch Tastensfeld mit LCD-Anzeige
- Preis/Leistungsverhältnis
- optional: Farbdruck, Erweiterung des Druckpuffers auf 40 kByte, automatischer Einzelblatteinzug

### Negativ:

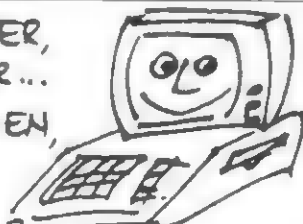
- Geräuschkulisse beim Drucken relativ hoch

WIR HABEN COMPUTER,  
ZUBEHÖR, LITERATUR...

ZUM SPIELEN, LERNEN,  
ARBEITEN...

BESUCHEN SIE UNS  
ODER FORDERN SIE KOSTENLOSE  
PREISLISTE AN!

B&C EDV-Systeme GmbH  
Tel.: 0222/5054978 Fax: 0222/5054978  
Mo-Fr 9<sup>00</sup>-12<sup>00</sup> u. 13<sup>00</sup>-18<sup>00</sup> Sa 9<sup>00</sup>-12<sup>00</sup>



## CompiMate

H. Rodat J. Haas M. Kiel

Der Computerladen Ihr AMIGA Spezialist in Ostwestfalen!

Festplattensysteme mit dem neuen TRUMPCARD - Controller

(getestet in AMIGA 01/02 90, mit Harddiskunterstützung für A-MAX)

TRUMPCARD A2000 SCSI-Controller einzeln DM 498,-

(Superschnell bis 500 KB/sec, Autoboot etc.)

als Filecard mit SEAGATE ST 157N (46 MB) DM 1328,-

oder Vollgas mit QUANTUM P40 S (40 MB) DM 1598,-

NEU für AMIGA 500 im Gehäuse, mit Steckplatz für Speichererweiterung max 4 MB

TRUMPCARD 500 5 MB komplett DM 1398,-

dto. natürlich auch mit QUANTUM P40S 40 MB DM 1648,-

Knaller: Hurricane für A500 mit 68020 / 16 MHz DM 1098,-

Hurricane H2800 68030/28 MHz f. A2000 DM 2398,-

Speichererweiterung A500 intern mit 2 MB best DM 649,-

Weitere aktuelle Angebote finden Sie in unserer Preisliste, die wir Ihnen gerne zusenden.  
CompiMate Computer, Sudbrackstr. 31, 31061 T. 0521-133621 / FAX 124 333



**U**nser Programm LinkChecker V2.0 (im folgenden kurz LC genannt) soll diese Aufgabe übernehmen. Das Programm benötigt eine ASCII-Datei, in der die zu überprüfenden Programme und deren Programmlänge angegeben sein müssen. Abbildung 1 zeigt, wie diese Batchdatei aussehen könnte.

Da es mehrere ASCII-Files auf Diskette geben kann, muß sich unser Parameterfile von den anderen Textfiles unterscheiden. Dies geschieht durch den Befehl LINK, der am Anfang des Parameterfiles stehen muß. Dabei braucht auf Groß- oder Kleinschreibung nicht geachtet zu werden.

## Am Anfang stand LIST

Will man beispielsweise den Amiga-DOS-Befehl 'List' aus dem Verzeichnis df0:c (Originallänge 8972 Bytes) vergleichen lassen, sollte man sich an folgende Syntax im Parameterfile halten:

df0:c/list 8972

Also zuerst den Filename der zu überprüfenden Datei, dann die Bytelänge. Will man mehrere oder vielleicht alle Files des Directories df0:c testen, kann man im Parameterfile folgendes anwenden:

```
\df0:c
list 8972
run 2324
install 1800
delete 5568
type 2112
cd 1908
```

Zuerst wird überprüft, ob das Verzeichnis auch wirklich existiert. Wenn nicht, passiert bis zum nächsten Directorywechsel gar nichts mehr, das heißt, alle folgenden Files werden einfach ignoriert.

```
LINK
\df0:devs
mountlist 75 ;Dies ist lediglich
DiskMaster.Config 706 ;ein Beispiel für
system-configuration 232 ;den Aufbau eines

\df0:libs
mathtrans.library 4280 ;Parameterfiles.
icon.library 5996
ConHandler.library 6152 ;Die hier angegebenen
translator.library 10592 ;File-Längen müssen
info.library 15744 ;nicht übereinstimmen!
mathseedoubbas.library 4352
version.library 400
diskfont.library 3968

\df0:
LinkChecker 2588

\df0:c
run 2324
setmap 4500
install 1800
delete 5568
type 2112
FailAt 1072
copy 6684
list 8972
cd 1908
```

Abb. 1. So könnte ein Parameterfile für den LinkChecker aussehen

Willi Bäcker

## LinkChecker V2.0

### Linkviren? – Mir doch egal!

Da es für einen Virus unmöglich sein sollte, ein Programm zu infizieren und dabei die Filelänge nicht zu verändern, könnte man doch auf den Gedanken kommen, ein Programm zu schreiben, das die tatsächliche Länge mit der in einer Batchdatei angegebenen Byte-Anzahl vergleicht. Somit dürften die gefährlichen Linkviren wohl keine Chance mehr haben.

Sollte es dieses Directory wider Erwarten doch geben, wird dort nach den aufgeführten Files gesucht. Man spart sich dadurch eine ganze Menge Tipparbeit.

Bei Bedarf können in dem Parameterfile auch Kommentare gesetzt werden, wie das folgende Beispiel zeigt:

```
\df0:c ; jetzt hab ich das
Directory gewechselt
```

list 8972 ; list hat 8972 Bytes

```
\df0: ; jetzt bin ich wieder imStammverzeichnis
```

Den letzten Directorywechsel im obigen Beispiel kann man sich natürlich sparen, denn sobald das komplette Parameterfile abgearbeitet ist, wird sofort wieder in das Verzeichnis gesprungen, in dem man sich ursprünglich befand.

Die Zeilen **müssen** immer mit RETURN abgeschlossen werden. Außer diesem RETURN wird jeder Fehler im Parameterfile übrigens dann bei der Ausführung gnadenlos angezeigt. Aufrufen kann man den LC mit:

### LinkChecker parameterfile

Hierbei wird nur eine kleine Message ausgegeben, und nur eventuell auftretende Abweichungen von Dateilängen werden angezeigt.

Man kann den LC aber auch mit verschiedenen Parametern aufrufen:

### LinkChecker parameterfile -s

Dabei steht der Parameter -s für show, man wird also über jede Aktion informiert. Ein weiterer Parameter liegt mit dem Zusatz -v vor. Gibt man diese Option mit an (v steht für Virus), so werden nun KickTagPtr, KickMemPtr, CoolCapture, ColdCapture und der OpenLibrary-Aufruf der exec.library überprüft. Beide Optionen lassen sich auch verknüpfen. Der Aufruf wäre dann folgendermaßen:

### LinkChecker parameterfile -sv

Wir brauchen sicherlich nicht zu erwähnen, daß unser Programm auch mit Fesplatten arbeitet – sonst wäre die Geschichte ja auch nur halb so interessant.

Sollte es tatsächlich vorkommen, daß der LC die Meldung "Filelength incorrect!" ausgibt, kann es durchaus sein, daß sich ein Linkvirus in diesem File breitgemacht hat. In diesem Fall wird beim Verlassen des LC der Return-Code (Register d0) auf den Wert Zehn gesetzt. Dies führt zum Abbruch in der Startup-Sequenz. Mit dem Befehl FAILAT 20 kann man diesen Abbruch jedoch übergehen. Das entsprechende File sollte man dann etwas genauer unter die Lupe nehmen.

(br)

## Listings

```
1: *****
2: * LinkChecker V2.0 *
3: * written by Willi Bäcker *
4: * (M) 1990 by DMV-Verlag *
5: * Sprache: Assembler (Seka) *
6: *****
7:
8: Open = -30
9: Close = -30 -6
10: Read = -30 -12
11: Write = -30 -18
12: Output = -30 -30
13: Seek = -30 -36
14: Lock = -30 -54
15: Unlock = -30 -60
16: CurrentDir = -30 -96
17: Examine = -30 -72
```

Listing: LinkChecker V2.0

```
18:
19: OpenLibrary = -30 -522
20: CloseLibrary = -30 -384
21: AllocMem = -30 -168
22: FreeMem = -30 -180
23: FindTask = -30 -264
24:
25: EXEC: equ 4
26: CSI: equ $9b
27:
28: start:
29: lea Path(pc),a1
30: andi.l #$ff,d0
31: subq.l #1,d0
32: move.l d0,Paralength
33: beq NoPara ;there is no Par
34: ameter
35: CoPara: cmp.b #$0a,(a0)
36: beq.s CReady
37: move.b (a0)+,(a1)+
```

Listing: LinkChecker V2.0



```

37:      dbra      d0,CoPara
38: CReady:    lea      path(pc),a0
39:      cmp.w     #$3f00,(a0)      ;is it only a ?
40:      beq       abouthelp
41:      lea      path(pc),a0
42:      cmp.l     #'info',(a0)
43:      printinfo
44:      lea      path(pc),a0
45:      cmp.l     #'',(a0)+
46:      bne.s     OptLoop0
47:      clr.b     -1(a0)
48:      cmp.l     #'-',(a0)+
49:      bne.l     illegaloption
50: OptLoop1:   cmp.l     #0,(a0)
51:      beq.s     optout
52:      cmp.l     #'s',(a0)
53:      beq.s     foundS
54:      cmp.l     #'v',(a0)
55:      beq.s     foundV
56:      bra      illegaloption
57: founds:    st       OptionFlag
58:      tst.b     (a0)+
59:      bra.s     OptLoop1
60: foundv:    st       OptionFlag1
61:      tst.b     (a0)+
62:      bra.s     OptLoop1
63: optout:    move.l   EXEC,a0
64:      move.l    $2a(a0),CoolCapture
65:      move.l    $2e(a0),CoolCapture

66:      move.l    $222(a0),KickMemPtr
67:      move.l    $226(a0),KickTagPtr
68:      move.l    -$196(a0),OpenPtr
69:      bsr      OpenLib
70:      tst.b     OptionFlag
71:      beq.s     no_clear
72:      move.l    #CTearText,d0
73:      bsr      DosPrint
74: No_Clear:  tst.b     OptionFlag1
75:      beq.s     Vir5
76:      tst.l     ColdCapture
77:      beq.s     Vir1
78:      move.l    #ColdCaptureText,d0
79:      bsr      DosPrint
80: Vir1:      tst.l     CoolCapture
81:      beq.s     Vir2
82:      move.l    #CoolCaptureText,d0
83:      bsr      DosPrint
84: Vir2:      tst.l     KickMemPtr
85:      beq.s     Vir3
86:      move.l    #KickMemPtrText,d0
87:      bsr      DosPrint
88: Vir3:      tst.l     KickTagPtr
89:      beq.s     Vir4
90:      move.l    #KickTagPtrText,d0
91:      bsr      DosPrint
92: Vir4:      move.l    OpenPtr,d0
93:      swap
94:      cmp.l     $ffc,d0
95:      beq.s     Vir5
96:      move.l    #OpenPtrText,d0
97:      bsr      DosPrint
98: Vir5:      move.l    DosBase(pc),a6
99:      move.l    #Path,d1
100:      move.l    #1005,d2
101:      jsr      Open(a6)      ;Open
102:      tst.l     d0
103:      beq       NoParaFile
104:      move.l    d0,ParaHandle
105:      move.l    d0,d1
106:      move.l    #TestBuffer,d2
107:      moveq     #4,d3
108:      jsr      Read(a6)
109:      lea      TestBuffer(pc),a0
110:      moveq     #3,d0
111: test1:     cmp.b     #'a',(a0)
112:      bhi.s     test1l
113:      add.b     #$20,(a0)+
114: test1l:    dbra      d0,test1
115:      lea      TestBuffer(pc),a0
116:      cmp.l     #'link',(a0)      ;Test File
117:      bne       IncorrectFile
118:      move.l    DosBase(pc),a6
119:      move.l    ParaHandle(pc),d1
120:      jsr      Close(a6)
121:      clr.l     ParaHandle
122:      move.l    DosBase(pc),a6
123:      move.l    #Path,d1
124:      move.l    #1005,d2
125:      jsr      Open(a6)      ;Open
126:      move.l    d0,ParaHandle
127:      move.l    DosBase(pc),a6
128:      move.l    ParaHandle(pc),d1
129:      moveq     #0,d2
130:      moveq     #1,d3
131:      jsr      Seek(a6)

132:      move.l    DosBase(pc),a6
133:      move.l    ParaHandle(pc),d1
134:      moveq     #0,d2
135:      moveq     #0,d3
136:      jsr      Seek(a6)
137:      move.l    d0,ParaFilelength
138:      move.l    DosBase(pc),a6
139:      move.l    ParaHandle(pc),d1
140:      moveq     #0,d2
141:      moveq     #-1,d3
142:      jsr      Seek(a6)
143:      move.l    EXEC,a6
144:      move.l    parafilelength(pc),d0
145:      addq      #8,d0
146:      move.l    #$10002,d1
147:      jsr      AllocMem(a6)
148:      tst.l     d0
149:      beq       MemoryError
150:      move.l    d0,memad0
151:      move.l    DosBase(pc),a6
152:      move.l    ParaHandle(pc),d1
153:      move.l    memad0,d2
154:      move.l    ParaFilelength,d3
155:      jsr      Read(a6)

```

Listing: LinkChecker V2.0

```

156:      move.l    DosBase(pc),a6
157:      move.l    ParaHandle(pc),d1
158:      jsr      Close(a6)
159:      move.l    #beginText,d0
160:      bsr      DosPrint
161:      tst.b     OptionFlag
162:      beq       noinfo
163:      move.l    #ParaText,d0
164:      bsr      DosPrint
165:      move.l    #Path,d0
166:      bsr      DosPrint
167:      move.l    #RetText,d0
168:      bsr      DosPrint
169:      move.l    memad0(pc),a0      ;get TextBuffer
170:      addq.l    #4,a0
171:      -----
172: doJob:      move.l    mytask(pc),a3
173:      btst     #12,28(a3)
174:      beq.s     huch
175:      move.l    #ExitText,d0
176:      bsr      DosPrint
177:      bra      Closeall
178: huch:      cmp.b     #0,(a0)      ;EndMark
179:      beq       Closeall
180:      cmp.b     #'!',(a0)+
181:      blo.s     doJob
182:      tst.b     -(a0)
183:      lea      NameBuffer(pc),a1
184:      cmp.b     #'',(a0)      ;followed by a D

185:      beq       Directory
186:      cmp.b     #'!',(a0)
187:      beq       Comment
188:      move.b     (a0)+,(a1)+
189:      cmp.b     #'!',(a0)
190:      bhs.s     doJob1
191:      lea      ZiffBuffer(pc),a1
192: doJob2:     cmp.b     #'!',(a0)+
193:      blo.s     doJob2
194:      tst.b     -(a0)
195: doJob3:     move.b     (a0)+,(a1)+
196:      cmp.b     #'!',(a0)

197:      bhs.s     doJob3
198:      move.l    a0,store0
199:      tst.b     DirFlag
200:      bmi       doJob
201:      lea      ZiffBuffer(pc),a4
202:      bsr      dec2bin      ;convert to hex
203:      tst.b     ZiffFlag
204:      bne       ValueError
205:      move.l    d0,Filelength
206:      tst.b     OptionFlag
207:      beq.s     nooptions
208:      move.l    d0-d7/a0-a6,-(sp)
209:      move.l    #workText,d0
210:      bsr      DosPrint
211:      move.l    #NameBuffer,d0
212:      bsr      DosPrint
213:      move.l    (sp)+,d0-d7/a0-a6
214: nooptions:  move.l    DosBase(pc),a6
215:      move.l    #NameBuffer,d1
216:      move.l    #1005,d2
217:      jsr      Open(a6)
218:      tst.l     d0
219:      beq       Jobnotfound
220:      move.l    d0,Filehandle
221:      move.l    DosBase(pc),a6
222:      move.l    FileHandle(pc),d1
223:      moveq     #0,d2
224:      moveq     #1,d3
225:      jsr      Seek(a6)
226:      move.l    DosBase(pc),a6
227:      move.l    FileHandle(pc),d1
228:      moveq     #0,d2
229:      moveq     #0,d3
230:      jsr      Seek(a6)
231:      move.l    d0,Filelength1
232:      move.l    DosBase(pc),a6
233:      move.l    FileHandle(pc),d1
234:      moveq     #0,d2
235:      moveq     #-1,d3
236:      jsr      Seek(a6)
237:      move.l    DosBase(pc),a6
238:      move.l    FileHandle(pc),d1
239:      jsr      Close(a6)
240:      move.l    filelength(pc),d0
241:      move.l    filelength1(pc),d1
242:      cmp.l     d0,d1
243:      bne       notcorrect
244:      clear:    lea      namebuffer(pc),a0
245:      moveq     #89,d0
246:      nf11:     clr.b     (a0)+
247:      dbra      d0,nf11
248:      move.l    store0,a0
249:      bra      doJob
250:      -----
251: Closeall:   tst.l     locker
252:      beq.s     nolocker
253:      move.l    locker,d1
254:      jsr      CurrentDir(a6)
255:      move.l    OldLock
256:      beq.s     nooldlock
257:      move.l    OldLock,d1
258:      jsr      CurrentDir(a6)
259:      move.l    #RetText,d0
260:      bsr      DosPrint
261:      move.l    EXEC,a6
262:      -----

263:      move.l    memad0(pc),a1
264:      move.l    ParaFilelength(pc),d0
265:      addq      #8,d0
266:      jsr      FreeMem(a6)
267:      move.l    #Cursor,d0
268:      bsr      DosPrint
269:      bsr      CloseLib
270:      moveq     #0,d0
271:      move.b     CLIFlag,d0
272:      sub.l     a0,a0
273:      rts

```

Listing: LinkChecker V2.0



```

274: ;-----
275: comment:      tst.b    (a0)+
276: comm1:        cmp.b    #10,(a0)+
277:              bne.s     comm1
278:              bra       dojob
279: ;-----
280: Directory:    bsr       clearDirBuffer
281:              lea       dirBuffer(pc),a1
282:              tst.b     (a0)+
283:              move.b    (a0)+,(a1)+
284:              cmp.b     #1,(a0)
285:              bhs.s     dojob
286:              movem.l   d0-d7/a0-a6,-(sp)
287:              tst.l     OldLock
288:              beq.s     no_OldLocks
289:              move.l    OldLock,d1
290:              move.l    DosBase,a6
291:              jsr       CurrentDir(a6)
292: no_OldLocks:   move.l    DosBase,a6
293:              move.l    #DirBuffer,d1
294:              moveq     #-2,d2
295:              jsr       Lock(a6)
296:              tst.l     d0
297:              beq       dirNotFound
298:              move.l    d0,locker      ;neuen DirLock h
299: ;-----
300: oten          move.l    d0,d1
301:              jsr       CurrentDir(a6)
302:              move.l    d0,OldLock    ;alten Lock spei
303: ;-----
304: chern         movem.l   (sp)+,d0-d7/a0-a6
305:              tst.b     OptionFlag
306:              beq       noDirOption
307:              move.l    #workText1,d0
308:              bsr       DosPrint
309:              move.l    #DirBuffer,d0
310:              bsr       DosPrint
311:              move.l    #RetText,d0
312:              bsr       DosPrint
313: noDirOption:  clr.b     DirFlag
314:              lea       NameBuffer(pc),a4
315:              moveq     #89,d0
316:              clr.b     (a4)+
317:              dbr       d0,dir5
318:              bra       dojob
319: ;-----
320: illegalOption: bsr       OpenLib
321:              move.l    #OptionText,d0
322:              bsr       DosPrint
323:              bsr       CloseLib
324:              moveq     #0,d0
325:              sub.l     a0,a0
326:              rts
327: ;-----
328: dirNotFound:  st        DirFlag
329: ;-----
330:              tst.b     OptionFlag
331:              beq.s     noDir7
332:              move.l    #RetText,d0
333:              bsr       DosPrint
334:              move.l    #NoDirText,d0
335:              bsr       DosPrint
336:              move.l    #DirBuffer,d0
337:              bsr       DosPrint
338:              move.l    #AsciiText2,d0
339:              bsr       DosPrint
340:              tst.b     OptionFlag
341:              bne.s     noDir3
342:              move.l    #RetText,d0
343:              bsr       DosPrint
344:              move.l    (sp)+,d0-d7/a0-a6
345:              bra       dojob
346: ;-----
347: clearDirBuffer: movem.l   d0/a0,-(sp)
348:              lea       dirBuffer(pc),a0
349:              moveq     #99,d0
350:              clr.b     (a0)+
351:              dbr       d0,clearloop
352:              movem.l   (sp)+,d0/a0
353:              rts
354: ;-----
355: clearloop:    dbr       d0,clearloop
356:              movem.l   (sp)+,d0/a0
357:              rts
358: ;-----
359: ValueError:   tst.b     OptionFlag
360:              beq.s     Value1
361:              move.l    #RetText,d0
362:              bsr       DosPrint
363:              move.l    #AsciiText,d0
364:              bsr       DosPrint
365:              move.l    #NameBuffer,d0
366:              bsr       DosPrint
367:              move.l    #AsciiText1,d0
368:              bsr       DosPrint
369:              tst.b     OptionFlag
370:              bne.s     Value2
371:              move.l    #RetText,d0
372:              bsr       DosPrint
373:              lea       NameBuffer(pc),a4
374:              moveq     #89,d0
375:              clr.b     (a4)+
376:              dbr       d0,vloop
377:              clr.b     ZiffFlag
378:              bra       dojob
379: ;-----
380: JobNotFound:  tst.b     OptionFlag
381:              bne.s     Job1
382:              move.l    #NameBuffer,d0
383:              bsr       DosPrint
384:              move.l    #FileNotFound,d0
385:              bsr       DosPrint
386:              move.l    #OptionFlag
387:              bne.s     Job2
388:              move.l    #RetText,d0
389:              bsr       DosPrint
390:              lea       NameBuffer(pc),a0
391:              moveq     #89,d0
392:              clr.b     (a0)+
393:              dbr       d0,nf1
394:              move.l    store0,a0
395:              move.b    #5,CLIFlag
396:              bra       dojob
397: ;-----
398: notCorrect:   tst.b     OptionFlag
399:              bne.s     not2

```

Listing: LinkChecker V2.0

```

393:              move.l    #NameBuffer,d0
394:              bsr       DosPrint
395: not2:         move.l    #inctext,d0
396:              bsr       DosPrint
397:              move.l    #RetText,d0
398:              bsr       DosPrint
399:              lea       NameBuffer(pc),a0
400:              moveq     #89,d0
401:              clr.b     (a0)+
402:              dbr       d0,nf12
403:              move.l    store0,a0
404:              move.b    #10,CLIFlag
405:              bra       dojob
406: ;-----
407: NoPara:       bsr       OpenLib      ;DosLib oeffnen
408:              move.l    #NoParaText,d0
409:              bsr       DosPrint
410:              bsr       CloseLib      ;DosLib schliess
411: ;-----
412: en           moveq     #0,d0
413:              sub.l     a0,a0
414:              rts
415: ;-----
416: NoParaFile:   move.l    #NoParaFText,d0
417:              bsr       DosPrint
418:              move.l    #Path,d0
419:              bsr       DosPrint
420:              move.l    #CheckText,d0
421:              bsr       DosPrint
422:              bsr       CloseLib      ;DosLib schliess
423: ;-----
424: en           moveq     #0,d0
425:              sub.l     a0,a0
426:              rts
427: ;-----
428: IncorrectFile: move.l    #IncorrectText,d0
429:              bsr       DosPrint
430:              move.l    DosBase(pc),a6
431:              move.l    ParaHandle(pc),d1
432:              jsr       Close(a6)
433:              bsr       CloseLib      ;DosLib schliess
434: ;-----
435: en           moveq     #0,d0
436:              sub.l     a0,a0
437:              rts
438: ;-----
439: MemoryError:  move.l    #MemErrorText,d0
440:              bsr       DosPrint
441:              move.l    DosBase(pc),a6
442:              move.l    ParaHandle(pc),d1
443:              jsr       Close(a6)
444:              bsr       CloseLib
445:              moveq     #0,d0
446:              sub.l     a0,a0
447:              rts
448: ;-----
449: LibError:     moveq     #4,d1
450:              moveq     #-1,d0
451:              move.w    d0,d2
452:              andi.l    #$0f00,d2
453:              move.w    d2,$dff180
454:              dbr       d0,Lib2
455:              dbr       d1,Lib1
456:              moveq     #1,d0
457:              sub.l     a0,a0
458:              rts
459: ;-----
460: OpenLib:      move.l    EXEC,a6
461:              sub.l     a1,a1
462:              jsr       FindTask(a6)
463:              move.l    d0,mytask
464:              lea       DosName(pc),a1
465:              moveq     #0,d0
466:              jsr       OpenLibrary(a6) ;OpenLibrary
467:              beq.s     LibError
468:              move.l    d0,DosBase    ;store DosBase
469:              move.l    d0,a6
470:              jsr       Output(a6)
471:              move.l    d0,CliHandle
472:              rts
473: ;-----
474: CloseLib:     move.l    EXEC,a6
475:              move.l    DosBase(pc),a1
476:              jsr       CloseLibrary(a6)
477:              rts
478: ;-----
479: DosPrint:     movem.l   d0-d3/a0/a6,-(sp) ;alle Regist
480:              er auf STACK
481:              move.l    d0,d2
482:              move.l    d2,d3
483:              move.l    d2,a0
484:              ;Stringadresse =
485:              ebergeben
486:              DosPrint0:  tst.b    (a0)+
487:              dresse + 1
488:              bne.s     DosPrint0
489:              ;Sprung wenn nic
490:              ht null
491:              sub.l     a0,d3
492:              neg.l     d3
493:              subq     #1,d3
494:              ;positiv machen.
495:              move.l    CliHandle,d1
496:              move.l    DosBase,a6
497:              jsr       Write(a6)
498:              ;Basisadr. holen
499:              ;(dos.library) =
500:              -48(a6)
501:              movem.l   (sp)+,d0-d3/a0/a6 ;Register vo
502:              m STACK holen
503:              rts
504:              ;zurueck
505: ;-----
506: aboutHelp:    bsr       openlib
507:              move.l    DosBase(pc),a6
508:              move.l    #AboutText,d0
509:              bsr       DosPrint
510:              bsr       CloseLib
511:              moveq     #0,d0
512:              rts
513: ;-----
514: PrintInfos:   bsr       openlib
515:              move.l    DosBase(pc),a6
516:              move.l    #InfoText,d0
517:              bsr       DosPrint
518:              bsr       CloseLib
519:              moveq     #0,d0
520:              rts

```

Listing: LinkChecker V2.0



```

506: -----
507: ;Dezimalstring to Binaer
508: ; a4 = Zeiger auf String
509: ; d0 = Ergebnisregister
510: Dec2Bin: movem.l d1-d7/a0-a6, -(sp)
511:         moveq #0, d0           ; Ergebnisregister
512:         moveq #0, d2           ; Naechste Ziffer
513: DecLoop: move.b (a4)+, d2      ; d2 = Zeichen aus
String:
514:         tst.b d2

515:         beq.s decout
516:         cmpi.b #'0', d2        ; Dezimalziffer?
517:         blo.s DecOut1
518:         cmpi.b #'9', d2
519:         bhi.s DecOut1         ; nein -->
520:         add.l d0, d0
521:         move.l d0, d1
522:         add.l d0, d0
523:         add.l d0, d0
524:         add.l d1, d0
525:         subi.b #$30, d2
526:         add.l d2, d0
527:         bra.s DecLoop
528: DecOut: subq.l #1, a4
529:         movem.l (sp)+, d1-d7/a0-a6
530:         rts
531: DecOut1: st
532:         bra.s DecOut
533: -----
534: ;Binaerzahl in Dezimal String wandeln
535: ;a5= Zeiger auf StringBuffer
536: ;Zahl steht in d4
537: BinDez: movem.l d0-d7/a0-a6, -(sp)
538:         tst.l d4              ; Zahl in d4
539:         beq.s BinDez6         ; branch if null
540:         bmi.s BinDez1         ; branch if Minus
541:         neg.l d4              ; neg it
542:         bra.s BinDez2
543: BinDez1: move.b #'-', (a5)+
544: BinDez2: lea Decathlon, a0
545:         clr.w d1
546: BinDez3: move.l (a0)+, d2
547:         beq.s BinDez6
548:         moveq #-1, d0
549: BinDez4: add.l d2, d4
550:         dbgt d0, BinDez4
551:         sub.l d2, d4
552:         addq.w #1, d0
553:         bne.s BinDez5
554:         tst.w d1
555:         beq.s BinDez3
556: BinDez5: moveq #-1, d1
557:         neg.b d0
558:         addi.b #$30, d0
559:         move.b d0, (a5)+
560:         bra.s BinDez3
561: BinDez6: neg.b d4
562:         addi.b #$30, d4
563:         move.b d4, (a5)+
564:         movem.l (sp)+, d0-d7/a0-a6
565:         rts
566: Decathlon:
567:         dc.l 1000000000
568:         dc.l 1000000000
569:         dc.l 1000000000
570:         dc.l 1000000000
571:         dc.l 1000000000
572:         dc.l 1000000000
573:         dc.l 1000000000
574:         dc.l 1000000000
575:         dc.l 1000000000
576:         dc.l 1000000000
577: -----
578: DosName: dc.b "dos.library", 0
580: ColdCapture: dc.l 0

581: CoolCapture: dc.l 0
582: KickMemPtr: dc.l 0
583: KickTagPtr: dc.l 0
584: OpenPtr: dc.l 0
585: DosBase: dc.l 0
586: CtlHandle: dc.l 0
587: store0: dc.l 0
588: store1: dc.l 0
589: memadd0: dc.l 0
590: ParaLength: dc.l 0
591: ParaFileLength: dc.l 0
592: ParaHandle: dc.l 0
593: OldLock: dc.l 0
594: Locker: dc.l 0
595: InfoLock: dc.l 0
596: Path: blk.b 80, 0
597: NameBuffer: blk.b 80, 0
598: ZiffBuffer: blk.b 10, 0
599: TestBuffer: dc.l 0
600: Filelength: dc.l 0 ;converted by AS
CITable:
601: Filelength1: dc.l 0 ;original length
602: FileHandle: dc.l 0
603: mytask: dc.l 0
604: ZiffFlag: dc.b 0, 0
605: CLIFlag: dc.b 0, 0
606: OptionFlag: dc.b 0, 0
607: OptionFlag1: dc.b 0, 0
608: DirFlag: dc.b 0, 0
609: DirBuffer: blk.b 100, 0
610: align 4
611: infoBlock: blk.b 80, 0
612: -----
613: ;Texts:
614: ReturnText: dc.b 10
615: RetText: dc.b 10, 0
616: ClearText: dc.b $c, 0
617: NoParaText: dc.b 10, 10
618:         dc.b "ERROR !", 10
619:         dc.b "USAGE: LinkChecker [FileName]"
620:         dc.b 10, 10, 0
621: -----
622: NoParaFText: dc.b 10, 10
623:         dc.b "ERROR !", 10

```

Listing: LinkChecker V2.0

```

624:         dc.b "Cannot find parameterfile : "
625:         dc.b 0
626: -----
627: MemErrorText: dc.b 10, 10
628:         dc.b "ERROR !", 10
629:         dc.b "Cannot allocate needed memorys
ize.", 10
630:         dc.b "Reset your AMIGA and try again
!", 10
631:         dc.b 0
632: -----
633: IncorrectText: dc.b 10, 10
634:         dc.b "ERROR !", 10
635:         dc.b "This is not a correct paramete
r file !", 10
636:         dc.b "Be sure that your parameter fi
le "
637:         dc.b "starts with 'LINK' !", 10
638:         dc.b 10, 0
639: -----
640: AboutText:
641:         dc.b 10, 10, 10

642:         dc.b "USAGE : ", 10
643:         dc.b "LinkCheckerV2.0 [File"
644:         dc.b "name] Options : [-sv]", 10
645:         dc.b "[v] = Virus+pointer chec"
646:         dc.b "k [s] = visual control", 10
647:         dc.b 10, 0
648: -----
649: incText: dc.b $d, CSI, "54C", CSI, "7;31;40m"
650:         dc.b "Filelength incorrect !"
651:         dc.b CSI, "0;31;40m"
652:         dc.b 0
653: -----
654: FileNotFound: dc.b $d, CSI, "54C", CSI, "7;31;40m"
655:         dc.b "File not found error !"
656:         dc.b CSI, "0;31;40m"
657:         dc.b 0
658: -----
659: NoDirText: dc.b CSI, "7;31;40m"
660:         dc.b "Directory not found >"
661:         dc.b CSI, "0;31;40m", 0
662: -----
663: AsciiText: dc.b CSI, "7;31;40m"
664:         dc.b "Illegal parameter at >"
665:         dc.b CSI, "0;31;40m", 0
666: -----
667: AsciiText1: dc.b $d, CSI, "54C", CSI, "7;31;40m"
668:         dc.b "Only DECIMAL numbers !"
669:         dc.b CSI, "0;31;40m", 0
670: -----
671: AsciiText2: dc.b $d, CSI, "54C", CSI, "7;31;40m"
672:         dc.b "Check parameterfile !"
673:         dc.b CSI, "0;31;40m", 0
674: -----
675: CheckText: dc.b 10, "Check: Drive/Path/Filename.
", 10, 0
676: -----
677: OptionText: dc.b 10
678:         dc.b "ERROR !! Illegal option recog
nized.",
679:         dc.b 10, 10, 0
680: -----
681: workText: dc.b $d, CSI, $4d
682:         dc.b "processing.....>"
683:         dc.b █
684: -----
685: workText1: dc.b 10, CSI, $4d
686:         dc.b "Directory changed to >"
687:         dc.b 0
688: -----
689: begintext: dc.b 10, CSI, $30, $20, $70
690:         dc.b "Linkchecker V2.0 running.", 10,
0
691: -----
692: ParaText: dc.b "Parameterfile: > ", 0
693: -----
694: ExitText: dc.b "C", 10
695: -----
696: Cursor: dc.b CSI, $20, $70, 0
697: -----
698: InfoText: dc.b 10, 10
699:         dc.b "*****"

700:         dc.b "LinkChecker V2.0", 10
701:         dc.b "written by Willi Baecker",
10
702:         dc.b "a FACTOR 5 production 1989"
, 10
703:         dc.b "*****"
704:         dc.b 0
705: ColdCaptureText: dc.b "WARNING !!! ColdCapture is pat
ched !!!"
706:         dc.b "Maybe a VIRUS in your Amiga
!", 10, 0
707:         dc.b "WARNING !!! CoolCapture is pat
ched !!!"
708:         dc.b "Maybe a VIRUS in your Amiga
!", 10, 0
709:         dc.b "WARNING !!! KickMemPointer is
patched !!!"
710:         dc.b "Maybe a VIRUS in your Amiga
!", 10, 0
711:         dc.b "WARNING !!! KickTagPointer is
patched !!!"
712:         dc.b "Maybe a VIRUS in your Amiga
!", 10, 0
713:         dc.b "WARNING !!! OpenLibrary is pat
ched !!!"
714:         dc.b "Maybe a LINK-VIRUS in your A
miga !", 10, 0
715:         dc.b ..
716:         align █
717:         dc.b "WARNING !!! OpenLibrary is pat
ched !!!"
718:         dc.b "Maybe a LINK-VIRUS in your A
miga !", 10, 0
719:         dc.b ..
720:         align █
721:         dc.b ..
722:         dc.b ..
723:

```

Listing: LinkChecker V2.0



Willi Bäcker

## LinkList

### Das Utility zum LinkChecker

Es ist sicherlich etwas umständlich im Umgang mit dem LinkChecker, sich alle Files, die überprüft werden sollen, zuerst listen zu lassen und dann mit der jeweiligen Programmlänge in ein Parameterfile zu bringen. Das Programm LinkList soll hier Abhilfe schaffen.

Das Programm LinkList gibt nach dem Aufruf alle Files eines Directories aus. Zusätzlich wird noch die Länge des Files und der Directoryname angegeben. Die Hauptaufgabe von LinkList besteht darin, die Parameterfiles für den LinkChecker

V2.0 einfach und schnell zu erweitern.

Es gibt mehrere Möglichkeiten, das Programm aufzurufen:

#### 1.) LinkList df0:

Hier wird zuerst der Directoryname auf dem Bildschirm ausgegeben (also \df0:), gefolgt von den Files, die im Stammverzeichnis von df0: enthalten sind.

#### 2.) LinkList > ram:test df0:

Wie bei Punkt 1, nur mit dem Unterschied, daß die Ausgabe nicht mehr auf den Bildschirm erfolgt, sondern in die Datei "ram:test" umgeleitet wird. Diese Datei kann mit einem beliebigen Editor verändert und für den LinkChecker bearbeitet werden.

#### 3.) LinkList df0:c df1:devs/ParameterFile

Der dritte Aufruf dieses Programms unterscheidet sich generell von den beiden anderen. In diesem Fall wird zuerst versucht, eine bereits vorhandene Parameterliste (hier df1:devs/ParameterFile) in den Speicher zu laden. Sollten dabei Probleme auftreten wie: 'File nicht vorhanden' oder 'Kein Parameterfile' (aller Wahrscheinlichkeit nach fehlt die Einleitung LINK), wird das Programm mit der entsprechenden Fehlermeldung abgebrochen. Ansonsten werden die Files (in unserem Beispiel aus df0:c) an die Parameterliste angehängt und die Batchdatei unter demselben Namen wieder zurückgeschrieben.

(br)

### Listings

```

1: *****
2: ***      LinkList      ***
3: ***      written by Willi Baecker      ***
4: ***      (c) 1990 by DMV-Verlag      ***
5: ***      Sprache: Assembler (Seka)      ***
6: *****
7:
8: Open      =      -30
9: Close     =      -30 -6
10: Read      =      -30 -12
11: Write     =      -30 -18
12: Output    =      -30 -30
13: Seek      =      -30 -36
14: Lock      =      -30 -54
15: Unlock    =      -30 -60
16: CurrentDir =      -30 -96
17: Examine   =      -30 -72
18: ExNext    =      -30 -78
19:
20: OpenLibrary =      -30 -522
21: CloseLibrary =      -30 -384
22: AllocMem   =      -30 -168
23: FreeMem    =      -30 -180
24: FindTask   =      -30 -264
25:
26: EXEC:      equ      4
27: CSI:       equ      $9b
28: Mode_old:  equ      1005
29: Mode_new:  equ      1006
30:
31: start:
32:   subq     #1,d0
33:   beq      usage      ;branch if no param
34:
35:   andi.l   $fff,d0      ;clear register
36:   move.b   d0,paralength ;store length
37:   cmp.w    #0,paralength
38:   beq      usage
39:   subq     #1,d0
40:   lea      path(pc),a1   ;for dbra
41:   loop1:   move.b   (a0)+,(a1)+ ;copy cliParameter t
42:   buffer
43:   cmp.b    #10,(a0)      ;RETURN found?
44:   beq      next          ;branch if next
45:   dbra     d0,loop1      ;next digit
46:   next:    lea      path(pc),a0 ;get buffer again
47:   moveq    #0,d0         ;clear register
48:   move.b   paralength,d0 ;get parameterlength
49:   loop2:   cmp.b    #1,(a0)+ ;is there an option?
50:   beq.s    found         ;Yes there is!
51:   dbra     d0,loop2      ;test next digit
52:   bra      nooption      ;no option
53:   found:   cmp.b    #10,(a0) ;correct option?
54:   beq      illegalOption ;no, go to hell!
55:   clr.b    -1(a0)        ;clear 1st comma
56:   move.l   a0,optionName ;store adress of Opt
57:   Name
58:   lea      OptionFbuffer(pc),a1
59:   ffl:     move.b   (a0)+,(a1)+
60:   tst.b    (a0)
61:   bne.s    ffl
62:   st       LinkFlag      ;set a Flag
63:   nooption:
64:   bar      Opndos        ;OpenLib + CliHandle
65:   tst.b    LinkFlag      ;was Option there?
66:   beq      directory    ;branch if not
67:
68:   move.l   DosBase(pc),a6 ;get Baseadress
69:   move.l   OptionName(pc),d1 ;get adress of Filen
70:
71:   move.l   #Mode_old,d2   ;I want to READ
72:   jsr      Open(a6)       ;Open file
73:   beq      Nooptionfile   ;branch on error
74:   move.l   d0,OptionHandle ;store handle
75:   move.l   DosBase(pc),a6
76:   move.l   OptionHandle(pc),d1
77:   #0,d2
78:   moveq    #1,d3
79:   jsr      Seek(a6)
80:
81:   move.l   d0,OptionFilelength
82:   move.l   DosBase(pc),a6
83:   move.l   OptionHandle(pc),d1
84:   moveq    #0,d2
85:   moveq    #-1,d3
86:   jsr      Seek(a6)
87:   EXEC,a6
88:   move.l   OptionFilelength(pc),d0
89:   add.l    #30000,d0
90:   move.l   #10002,d1
91:   jsr      AllocMem(a6)
92:   beq      MemoryError
93:   move.l   d0,memad0
94:   move.l   DosBase(pc),a6
95:   move.l   OptionHandle(pc),d1
96:   move.l   memad0,d2
97:   move.l   OptionFilelength,d3
98:   jsr      Read(a6)
99:   move.l   DosBase(pc),a6
100:  move.l   OptionHandle(pc),d1
101:  jsr      Close(a6)
102:  move.l   memad0(pc),a0
103:  moveq    #3,d0
104:  test1:   cmp.b    #'a',(a0)
105:  bhi.s    test1l
106:  add.b    #520,(a0)+
107:  test1l:  dbra     d0,test1
108:  move.l   memad0(pc),a0
109:  cmp.l    #'link',(a0) ;Test File
110:  bne      IncorrectFile
111:  move.l   memad0(pc),d0
112:  add.l    OptionFilelength,d0
113:  move.l   d0,newbuffer
114:  directory:
115:  move.l   DosBase(pc),a6 ;get LibBass
116:  move.l   #path,d1 ;get adress of Dirn
117:
118:  moveq    #-2,d2 ;mode: read
119:  jsr      Lock(a6)
120:  beq      error ;branch on error
121:  move.l   d0,locksave ;store new Lock
122:  move.l   DosBase(pc),a6
123:  move.l   locksave(pc),d1
124:  move.l   #fileinfo,d2
125:  jsr      Examine(a6)
126:  beq      error ;ready
127:  lea      fileinfo+4(pc),a0
128:  cmp.l    #-3,(a0) ;is it a File?
129:  beq      Fileerror ;Yes, branch!
130:
131:  move.l   #Return,d0
132:  bsr      DosPrint
133:  move.l   #slash,d0
134:  bsr      DosPrint
135:  move.l   #path,d0
136:  bsr      DosPrint
137:  move.l   #Return,d0
138:  bsr      DosPrint
139:  tst.b    LinkFlag
140:  beq      loop ;never trash a3
141:  move.l   newbuffer(pc),a3
142:  move.b   #10,(a3)+
143:  move.b   #'',(a3)+
144:  lea      path(pc),a0
145:  copy:    move.b   (a0)+,(a3)+
146:  cmp.b    #0,(a0)
147:  bne.s    copy
148:  move.b   #10,(a3)+
149:  loop:    move.l   mytask(pc),a4
150:  btst     #12,28(a4) ;Control C
151:  beq.s    control
152:  move.l   #ExitText,d0
153:  bsr      DosPrint
154:  bra      Error
155:  control: lea      21ffbuffer(pc),a0
156:  clr.l    (a0)+

```

Listing: LinkList

```

75:   move.l   DosBase(pc),a6
76:   move.l   OptionHandle(pc),d1
77:   moveq    #0,d2
78:   moveq    #0,d3
79:   jsr      Seek(a6)
80:   move.l   d0,OptionFilelength
81:   move.l   DosBase(pc),a6
82:   move.l   OptionHandle(pc),d1
83:   moveq    #0,d2
84:   moveq    #-1,d3
85:   jsr      Seek(a6)
86:   EXEC,a6
87:   move.l   OptionFilelength(pc),d0
88:   add.l    #30000,d0
89:   move.l   #10002,d1
90:   jsr      AllocMem(a6)
91:   beq      MemoryError
92:   move.l   d0,memad0
93:   move.l   DosBase(pc),a6
94:   move.l   OptionHandle(pc),d1
95:   move.l   memad0,d2
96:   move.l   OptionFilelength,d3
97:   jsr      Read(a6)
98:   move.l   DosBase(pc),a6
99:   move.l   OptionHandle(pc),d1
100:  jsr      Close(a6)
101:  move.l   memad0(pc),a0
102:  moveq    #3,d0
103:  test1:   cmp.b    #'a',(a0)
104:  bhi.s    test1l
105:  add.b    #520,(a0)+
106:  test1l:  dbra     d0,test1
107:  move.l   memad0(pc),a0
108:  cmp.l    #'link',(a0) ;Test File
109:  bne      IncorrectFile
110:  move.l   memad0(pc),d0
111:  add.l    OptionFilelength,d0
112:  move.l   d0,newbuffer
113:  directory:
114:  move.l   DosBase(pc),a6 ;get LibBass
115:  move.l   #path,d1 ;get adress of Dirn
116:
117:  moveq    #-2,d2 ;mode: read
118:  jsr      Lock(a6)
119:  beq      error ;branch on error
120:  move.l   d0,locksave ;store new Lock
121:  move.l   DosBase(pc),a6
122:  move.l   locksave(pc),d1
123:  move.l   #fileinfo,d2
124:  jsr      Examine(a6)
125:  beq      error ;ready
126:  lea      fileinfo+4(pc),a0
127:  cmp.l    #-3,(a0) ;is it a File?
128:  beq      Fileerror ;Yes, branch!
129:
130:  move.l   #Return,d0
131:  bsr      DosPrint
132:  move.l   #slash,d0
133:  bsr      DosPrint
134:  move.l   #path,d0
135:  bsr      DosPrint
136:  move.l   #Return,d0
137:  bsr      DosPrint
138:  tst.b    LinkFlag
139:  beq      loop ;never trash a3
140:  move.l   newbuffer(pc),a3
141:  move.b   #10,(a3)+
142:  move.b   #'',(a3)+
143:  lea      path(pc),a0
144:  copy:    move.b   (a0)+,(a3)+
145:  cmp.b    #0,(a0)
146:  bne.s    copy
147:  move.b   #10,(a3)+
148:  loop:    move.l   mytask(pc),a4
149:  btst     #12,28(a4) ;Control C
150:  beq.s    control
151:  move.l   #ExitText,d0
152:  bsr      DosPrint
153:  bra      Error
154:  control: lea      21ffbuffer(pc),a0
155:  clr.l    (a0)+

```

Listing: LinkList



```

155:      clr.l      (a0)+
156:      move.l     DosBase(pc),a6
157:      move.l     locksave(pc),d1
158:      move.l     #fileinfo,d2
159:      jsr        ExNext(a6)
160:      tst.l      d0
161:      beq        nofilemore
162:      lea        fileinfo(pc),a0
163:      move.l     120(a0),d3
164:      cmp.b      #3,d3
165:      bne        noz
166:      move.l     #fileinfo+8,d0
167:      bsr        dosprint
168:
169:      tst.b      LinkFlag
170:      beq        copy1
171:      move.l     #fileinfo+8,a0
172:      copy2:     move.b      (a0)+,(a3)+
173:      cmp.b      #0,(a0)
174:      bne.s      copy2
175:
176:      copy1:     move.l     d0,a0
177:      move.l     d0,d1
178:      nameloop:
179:      addq       #1,d0
180:      tst.b      (a0)+
181:      bne.s      nameloop
182:      sub.l      d1,d0
183:      subq       #1,d0
184:      moveq      #0,d2
185:      moveq      #33,d1
186:      sub.l      d0,d1
187:      divu       #8,d1
188:      move.w     d1,d2 ;d2 = tabs
189:      lea        textbuffer(pc),a0
190:      swap       d1
191:      tst.w      d1
192:      beq.s      nospace
193:      andi.l     #$ffff,d1
194:
195:      subq       #1,d1
196:
197:      sploop:    lea        space(pc),a1
198:      move.b      (a1),(a0)+
199:      dbra        d1,sploop
200:      nospace:
201:      tst.w      d2
202:      beq.s      notabs
203:      subq       #1,d2
204:      tabloop:  lea        tab(pc),a1
205:      move.b      (a1),(a0)+
206:      dbra        d2,tabloop
207:      notabs:
208:      move.l     #textbuffer,d0
209:      bsr        DosPrint
210:
211:      test:      tst.b      LinkFlag
212:      beq.s      copy4
213:
214:      copy3:     lea        Textbuffer(pc),a0
215:      move.b      (a0)+,(a3)+
216:      cmp.b      #0,(a0)
217:      bne.s      copy3
218:
219:      copy4:     lea        fileinfo(pc),a0
220:      move.l     124(a0),d4
221:      tst.l      d4
222:      beq        noziffer
223:      lea        ziffbuffer(pc),a5
224:      bsr        bindez
225:
226:      move.l     #ziffbuffer,d0
227:      bsr        dosprint
228:
229:      tst.b      LinkFlag
230:      beq        noziffer
231:      lea        ziffbuffer(pc),a0
232:      copy6:     move.b      (a0)+,(a3)+
233:      cmp.b      #0,(a0)
234:      bne.s      copy6
235:      move.l     #10,(a3)+
236:      move.l     a3,Endefile
237:
238:      noziffer:
239:      move.l     #return,d0
240:      bsr        dosprint
241:
242:      noz:
243:      lea        Textbuffer(pc),a0
244:      moveq      #79,d0
245:      (a0)+
246:      d0,clrbuff
247:      dbra        d0,clrbuff
248:
249:      bra        loop
250:
251:      error:
252:      tst.l      memad0
253:      beq.s      noFreeMem
254:      move.l     EXEC,a6
255:      move.l     memad0(pc),a1
256:      move.l     OptionFilelength,d0
257:      add.l      #30000,d0
258:      jsr        FreeMem(a6)
259:
260:      noFreeMem:
261:      move.l     #return,d0
262:      bsr        dosprint
263:      bsr        CloseDos
264:
265:      noparas:
266:      moveq      #0,d0
267:
268:      sub.l      a0,a0
269:      rts
270:
271:      ;
272:      nofilemore:
273:      tst.b      LinkFlag
274:      beq.s      Error
275:
276:      move.l     DosBase(pc),a6
277:      move.l     #OptionFBuffer,d1
278:      move.l     #mode_new,d2
279:      jsr        Open(a6)
280:      move.l     d0,writehandle
281:      move.l     d0,d1
282:      move.l     endefile,d3
283:      move.l     memad0,d2
284:      sub.l      d2,d3

```

Listing: LinkList

Amiga 2000 Memory 2 MB	6.490,-
Amiga 2000 Harddisk GVP 80 MB/19 ms	21.990,-
Amiga 2000 68030 Karte von GVP/25 MHz	17.490,-
Amiga 2000 LIVE Digitizer (Realtime)	10.490,-
Amiga 2000 Genlock's	
Amiga 500 Hurrican 68020/68881, 1 MB Fastram	
32 Bits Karte inkl. Prozessor	15.990,-
Amiga 500 Memory 512 KB mit Uhr	1.590,-
Amiga Pal Genlock 1.3 + RGB Splitter	4.890,-
Amiga S-VHS Genlock + RGB Splitter	8.990,-
Amiga Laufwerk 3.5/Bus/Ein-, Ausschalter	1.990,-
Amiga Laufwerk 5.25, sonst wie oben + 40/80 Tr.	2.690,-
Amiga Ersatz Maus, mit Microschalter, Mausepad	749,-
Midi-Interface	799,-
Midi-Interface + Midi Software	999,-
Soundsampler + Software (Stereo)	1.390,-
Scanner 10,5 cm/400 DPI	4.890,-

#### Software:

Audiomaster II	999,-
Deluxe Video 1.2 Deutsch	890,-
Pagestream 1.8	3.490,-
Go Amiga Text + Datei	399,-
Hires Workbench	329,-
Digi View 4.0	2.490,-
Digi Paint III	1.190,-

Alle Gold-Disk Programme zu Super Preisen

Große Auswahl an Zubehör und Büchern lagernd!!!

Alle Preise in öS inkl. 20 % MWST., Druckfehler und Preisänderungen vorbehalten

## Spiele für Amiga

### SPEICHERERWEITERUNG AMIGA 500

auf 1 MB, inklusive Uhr, abschaltbar 178,00DM

mit dem Spiel DUNGEON MASTER (dtsh.) 248,00DM

Wir bekommen täglich Neuheiten für den Amiga! Fordern Sie doch einfach mal die neueste Preisliste an, Sie werden überrascht sein! Hier ein paar Beispiele:

NORDIC POWER	SOFTWARE-TITEL	PREIS
	Aquanaut	74,80
Der Freezer für den	Fighter Bomber	89,80
Amiga 500!	It came from the Desert	84,80
	Kaiser	109,00
Fordern Sie ausführliche	Little Computer People	19,80
Unterlagen an!	Treasure Island Dizzy	19,80
	Twinworld	74,80
	Space Harrier II	59,90

X-Copy II + Cyclone + Hardware nur 69,00DM

#### Versandkosten:

5,00DM bei Warenwert unter 100,00DM

Warenwert über 100,00DM Versandkosten frei!

## COMPY/SHOP

Gneisenastr. 29  
4330 Mülheim Ruhr  
0208-497169/496178

```

276:      jsr      Write(a6)
277:      bmi      writeError
278: back:  move.l  writehandle,d1
279:      jsr      Close(a6)
280:      bra      error
281:
282: writeError:
283:      move.l  #writeErrText,d0
284:      bsr     DosPrint
285:      bra      back
286:
287: -----
288: usage:
289:      bsr     OpenDos
290:      move.l  #UsageText,d0
291:      bsr     DosPrint
292:      bsr     CloseDos
293:      moveq   #0,d0
294:      sub.l   a0,a0
295:      rts
296:
297: -----
298: illegalOption:
299:      bsr     OpenDos
300:      move.l  #IllegalText,d0
301:      bsr     DosPrint
302:      bsr     CloseDos
303:      sub.l   a0,a0
304:      moveq   #0,d0
305:      rts
306:
307: -----
308: incorrectFile:
309:      move.l  #IncorrectText,d0
310:      bsr     DosPrint
311:      bra      Error
312:
313: -----
314: fileError:
315:      move.l  #FileErrorText,d0
316:      bsr     DosPrint
317:      bra      Error
318:
319: -----
320: memoryError:
321:      move.l  #MemErrorText,d0
322:      bsr     DosPrint
323:      move.l  DosBase(pc),a6
324:      move.l  OptionHandle(pc),d1
325:      jsr     Close(a6)
326:      bsr     CloseDos
327:      moveq   #0,d0
328:      sub.l   a0,a0
329:      rts
330:
331: -----
332: nooptionFile:
333:      move.l  #OptionFileError,d0
334:      bsr     DosPrint
335:      move.l  OptionName(pc),d0
336:      bsr     DosPrint
337:      move.l  #Return,d0
338:      bsr     DosPrint
339:      bsr     CloseDos
340:      sub.l   a0,a0
341:      moveq   #0,d0
342:      rts
343:
344: -----
345: openDos:
346:      move.l  4,a6
347:      sub.l   a1,a1
348:      jsr     FindTask(a6)
349:      move.l  d0,mytask
350:      move.l  4,a6
351:      lea     DosName(pc),a1
352:      moveq   #0,d0
353:      jsr     OpenLibrary(a6)
354:      move.l  d0,DosBase
355:      move.l  d0,a6
356:      jsr     Output(a6)
357:      move.l  d0,CliHandle
358:      rts
359:
360: -----
361: closeDos:
362:      move.l  4,a6
363:      move.l  DosBase(pc),a1
364:      jsr     CloseLibrary(a6)
365:      rts
366:
367: -----
368: dosPrint:
369:      movem.l d0-d3/a0/a6,-(sp) ; alle Register auf S
370:      tack
371:      move.l  d0,d2 ; start retten
372:      move.l  d2,d3 ; String-adresse
373:      move.l  d2,a0 ; uebergeben
374:      tst.b   (a0)+ ; Byte testen & Adres
375:      bne.s   DosPrint0 ; Sprung wenn nicht n
376:      ull
377:      sub.l   a0,d3 ; positiv machen
378:      neg.l   d3
379:      subq    #1,d3
380:      move.l  CliHandle(pc),d1
381:      move.l  DosBase,a6 ; Basisadresse holen
382:      jsr     Write(a6) ; (dos.library) = -48
383:      (a6)
384:      movem.l (sp)+,d0-d3/a0/a6 ; Register vom STACK
385:      holen
386:      rts ; zurueck
387:
388: -----
389: ; Binaerzahl in DezimalString wandeln
390: ; a5= Zeiger auf StringBuffer
391: ; Zahl steht in d4
392: BinDez: movem.l d0-d7/a0-a6,-(sp)
393:      tst.l   d4 ; Zahl in d4
394:      beq.s   BinDez6 ; branch if null
395:      bmi.s   BinDez1 ; branch if minus
396:
397:      neg.l   d4 ; neg it
398:      bra.s   BinDez2
399: BinDez1: move.b #'-',(a5)+

```

Listing: LinkList

```

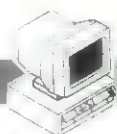
389: BinDez2: lea     Decathlon,a0
390:      clr.w   d1
391: BinDez3: move.l  (a0)+,d2
392:      beq.s   BinDez6
393:      moveq   #-1,d0
394: BinDez4: add.l   d2,d4
395:      dbgt    d0,BinDez4
396:      sub.l   d2,d4
397:      addq.w   #1,d0
398:      bne.s   BinDez5
399:      tst.w   d1
400:      beq.s   BinDez3
401: BinDez5: moveq   #-1,d1
402:      neg.b   d0
403:      addi.b   #530,d0
404:      move.b   d0,(a5)+
405:      bra.s   BinDez3
406: BinDez6: neg.b   d4
407:      addi.b   #530,d4
408:      move.b   d4,(a5)+
409:      movem.l  (sp)+,d0-d7/a0-a6
410:      rts
411:
412: -----
413: Decathlon:
414:      dc.l    1000000000
415:      dc.l    1000000000
416:      dc.l    1000000000
417:      dc.l    1000000000
418:      dc.l    100000
419:      dc.l    10000
420:      dc.l    100
421:      dc.l    10
422:      dc.l    0
423:
424: align 4
425: fileInfo:      blk.l 260,0
426: path:          blk.b 200,0
427: ziffbuffer:    dc.l 0,0,0,0
428: textbuffer:    blk.b 80,0
429: OptionBuffer:  blk.b 80,0
430:
431: locksave:      dc.l 0
432: DosBase:       dc.l 0
433: memad0:        dc.l 0
434: writehandle:   dc.l 0
435: clihandle:     dc.l 0
436: OptionHandle:  dc.l 0
437: OptionFilelength: dc.l 0
438: OptionName:    dc.l 0 ; address of OptFile na
439: newbuffer:     dc.l 0
440: EndeFile:      dc.l 0
441: mytask:        dc.l 0
442: parlength:     dc.b 0,0 ; len of parameters
443: LinkFlag:      dc.b 0,0
444:
445: DosName:       dc.b "dos.library",0
446:
447: Return:        dc.b 10,0
448:
449: slash:         dc.b "\",0
450:
451: space:         dc.b 32,0
452:
453: tab:           dc.b 9,0
454:
455: illegaltext:   dc.b 10
456:                dc.b "ERROR !",10
457:                dc.b "Illegal option recognized !",
458:                dc.b 0
459:
460: OptionFileError: dc.b 10
461:                dc.b "ERROR !",10
462:                dc.b "Unable to open Parameterfile"
463:                dc.b 0
464:
465: MemErrorText:  dc.b 10,10
466:                dc.b "ERROR !",10
467:                dc.b "Cannot allocate needed memory"
468:                dc.b 0
469:
470: size:          dc.b 10
471:                dc.b "Reset your AMIGA and try agai
472:                dc.b 0
473:
474: n!:            dc.b 10,10
475:                dc.b "ERROR !",10
476:                dc.b "Cannot write parameterfile!",
477:                dc.b 0
478:
479: UsageText:     dc.b "*****"
480:                dc.b "*****",10
481:                dc.b "Usage: LinkList DFx: [ Paramet
482:                dc.b "erfilename]",10
483:                dc.b "*****",10
484:                dc.b "*****",10
485:                dc.b "A FACTOR 5 production 1989 by
486:                dc.b "W.Baecker!",10
487:                dc.b "*****",10
488:                dc.b "*****",10
489:                dc.b "*****",10
490:                dc.b "*****",10
491:                dc.b "*****",10
492:                dc.b "*****",10
493:                dc.b "*****",10
494:                dc.b "*****",10
495:                dc.b "*****",10
496:                dc.b "*****",10
497:                dc.b "*****",10
498:                dc.b "*****",10
499:                dc.b "*****",10
500:                dc.b "*****",10
501:                dc.b "*****",10
502:                dc.b "*****",10
503:                dc.b "*****",10
504:                dc.b "*****",10
505:                dc.b "*****",10
506:                dc.b "*****",10
507:                dc.b "*****",10
508:                dc.b "*****",10
509:                dc.b "*****",10
510:                dc.b "*****",10
511:                dc.b "*****",10
512:                dc.b "*****",10
513:                dc.b "*****",10
514:                dc.b "*****",10
515:                dc.b "*****",10
516:                dc.b "*****",10
517:                dc.b "*****",10
518:                dc.b "*****",10
519:                dc.b "*****",10
520:                dc.b "*****",10
521:                dc.b "*****",10
522:                dc.b "*****",10
523:                dc.b "*****",10
524:                dc.b "*****",10
525:                dc.b "*****",10
526:                dc.b "*****",10
527:                dc.b "*****",10
528:                dc.b "*****",10
529:                dc.b "*****",10
530:                dc.b "*****",10
531:                dc.b "*****",10
532:                dc.b "*****",10
533:                dc.b "*****",10
534:                dc.b "*****",10
535:                dc.b "*****",10
536:                dc.b "*****",10
537:                dc.b "*****",10
538:                dc.b "*****",10
539:                dc.b "*****",10
540:                dc.b "*****",10
541:                dc.b "*****",10
542:                dc.b "*****",10
543:                dc.b "*****",10
544:                dc.b "*****",10
545:                dc.b "*****",10
546:                dc.b "*****",10
547:                dc.b "*****",10
548:                dc.b "*****",10
549:                dc.b "*****",10
550:                dc.b "*****",10
551:                dc.b "*****",10
552:                dc.b "*****",10
553:                dc.b "*****",10
554:                dc.b "*****",10
555:                dc.b "*****",10
556:                dc.b "*****",10
557:                dc.b "*****",10
558:                dc.b "*****",10
559:                dc.b "*****",10
560:                dc.b "*****",10
561:                dc.b "*****",10
562:                dc.b "*****",10
563:                dc.b "*****",10
564:                dc.b "*****",10
565:                dc.b "*****",10
566:                dc.b "*****",10
567:                dc.b "*****",10
568:                dc.b "*****",10
569:                dc.b "*****",10
570:                dc.b "*****",10
571:                dc.b "*****",10
572:                dc.b "*****",10
573:                dc.b "*****",10
574:                dc.b "*****",10
575:                dc.b "*****",10
576:                dc.b "*****",10
577:                dc.b "*****",10
578:                dc.b "*****",10
579:                dc.b "*****",10
580:                dc.b "*****",10
581:                dc.b "*****",10
582:                dc.b "*****",10
583:                dc.b "*****",10
584:                dc.b "*****",10
585:                dc.b "*****",10
586:                dc.b "*****",10
587:                dc.b "*****",10
588:                dc.b "*****",10
589:                dc.b "*****",10
590:                dc.b "*****",10
591:                dc.b "*****",10
592:                dc.b "*****",10
593:                dc.b "*****",10
594:                dc.b "*****",10
595:                dc.b "*****",10
596:                dc.b "*****",10
597:                dc.b "*****",10
598:                dc.b "*****",10
599:                dc.b "*****",10
600:                dc.b "*****",10
601:                dc.b "*****",10
602:                dc.b "*****",10
603:                dc.b "*****",10
604:                dc.b "*****",10
605:                dc.b "*****",10
606:                dc.b "*****",10
607:                dc.b "*****",10
608:                dc.b "*****",10
609:                dc.b "*****",10
610:                dc.b "*****",10
611:                dc.b "*****",10
612:                dc.b "*****",10
613:                dc.b "*****",10
614:                dc.b "*****",10
615:                dc.b "*****",10
616:                dc.b "*****",10
617:                dc.b "*****",10
618:                dc.b "*****",10
619:                dc.b "*****",10
620:                dc.b "*****",10
621:                dc.b "*****",10
622:                dc.b "*****",10
623:                dc.b "*****",10
624:                dc.b "*****",10
625:                dc.b "*****",10
626:                dc.b "*****",10
627:                dc.b "*****",10
628:                dc.b "*****",10
629:                dc.b "*****",10
630:                dc.b "*****",10
631:                dc.b "*****",10
632:                dc.b "*****",10
633:                dc.b "*****",10
634:                dc.b "*****",10
635:                dc.b "*****",10
636:                dc.b "*****",10
637:                dc.b "*****",10
638:                dc.b "*****",10
639:                dc.b "*****",10
640:                dc.b "*****",10
641:                dc.b "*****",10
642:                dc.b "*****",10
643:                dc.b "*****",10
644:                dc.b "*****",10
645:                dc.b "*****",10
646:                dc.b "*****",10
647:                dc.b "*****",10
648:                dc.b "*****",10
649:                dc.b "*****",10
650:                dc.b "*****",10
651:                dc.b "*****",10
652:                dc.b "*****",10
653:                dc.b "*****",10
654:                dc.b "*****",10
655:                dc.b "*****",10
656:                dc.b "*****",10
657:                dc.b "*****",10
658:                dc.b "*****",10
659:                dc.b "*****",10
660:                dc.b "*****",10
661:                dc.b "*****",10
662:                dc.b "*****",10
663:                dc.b "*****",10
664:                dc.b "*****",10
665:                dc.b "*****",10
666:                dc.b "*****",10
667:                dc.b "*****",10
668:                dc.b "*****",10
669:                dc.b "*****",10
670:                dc.b "*****",10
671:                dc.b "*****",10
672:                dc.b "*****",10
673:                dc.b "*****",10
674:                dc.b "*****",10
675:                dc.b "*****",10
676:                dc.b "*****",10
677:                dc.b "*****",10
678:                dc.b "*****",10
679:                dc.b "*****",10
680:                dc.b "*****",10
681:                dc.b "*****",10
682:                dc.b "*****",10
683:                dc.b "*****",10
684:                dc.b "*****",10
685:                dc.b "*****",10
686:                dc.b "*****",10
687:                dc.b "*****",10
688:                dc.b "*****",10
689:                dc.b "*****",10
690:                dc.b "*****",10
691:                dc.b "*****",10
692:                dc.b "*****",10
693:                dc.b "*****",10
694:                dc.b "*****",10
695:                dc.b "*****",10
696:                dc.b "*****",10
697:                dc.b "*****",10
698:                dc.b "*****",10
699:                dc.b "*****",10
700:                dc.b "*****",10
701:                dc.b "*****",10
702:                dc.b "*****",10
703:                dc.b "*****",10
704:                dc.b "*****",10
705:                dc.b "*****",10
706:                dc.b "*****",10
707:                dc.b "*****",10
708:                dc.b "*****",10
709:                dc.b "*****",10
710:                dc.b "*****",10
711:                dc.b "*****",10
712:                dc.b "*****",10
713:                dc.b "*****",10
714:                dc.b "*****",10
715:                dc.b "*****",10
716:                dc.b "*****",10
717:                dc.b "*****",10
718:                dc.b "*****",10
719:                dc.b "*****",10
720:                dc.b "*****",10
721:                dc.b "*****",10
722:                dc.b "*****",10
723:                dc.b "*****",10
724:                dc.b "*****",10
725:                dc.b "*****",10
726:                dc.b "*****",10
727:                dc.b "*****",10
728:                dc.b "*****",10
729:                dc.b "*****",10
730:                dc.b "*****",10
731:                dc.b "*****",10
732:                dc.b "*****",10
733:                dc.b "*****",10
734:                dc.b "*****",10
735:                dc.b "*****",10
736:                dc.b "*****",10
737:                dc.b "*****",10
738:                dc.b "*****",10
739:                dc.b "*****",10
740:                dc.b "*****",10
741:                dc.b "*****",10
742:                dc.b "*****",10
743:                dc.b "*****",10
744:                dc.b "*****",10
745:                dc.b "*****",10
746:                dc.b "*****",10
747:                dc.b "*****",10
748:                dc.b "*****",10
749:                dc.b "*****",10
750:                dc.b "*****",10
751:                dc.b "*****",10
752:                dc.b "*****",10
753:                dc.b "*****",10
754:                dc.b "*****",10
755:                dc.b "*****",10
756:                dc.b "*****",10
757:                dc.b "*****",10
758:                dc.b "*****",10
759:                dc.b "*****",10
760:                dc.b "*****",10
761:                dc.b "*****",10
762:                dc.b "*****",10
763:                dc.b "*****",10
764:                dc.b "*****",10
765:                dc.b "*****",10
766:                dc.b "*****",10
767:                dc.b "*****",10
768:                dc.b "*****",10
769:                dc.b "*****",10
770:                dc.b "*****",10
771:                dc.b "*****",10
772:                dc.b "*****",10
773:                dc.b "*****",10
774:                dc.b "*****",10
775:                dc.b "*****",10
776:                dc.b "*****",10
777:                dc.b "*****",10
778:                dc.b "*****",10
779:                dc.b "*****",10
780:                dc.b "*****",10
781:                dc.b "*****",10
782:                dc.b "*****",10
783:                dc.b "*****",10
784:                dc.b "*****",10
785:                dc.b "*****",10
786:                dc.b "*****",10
787:                dc.b "*****",10
788:                dc.b "*****",10
789:                dc.b "*****",10
790:                dc.b "*****",10
791:                dc.b "*****",10
792:                dc.b "*****",10
793:                dc.b "*****",10
794:                dc.b "*****",10
795:                dc.b "*****",10
796:                dc.b "*****",10
797:                dc.b "*****",10
798:                dc.b "*****",10
799:                dc.b "*****",10
800:                dc.b "*****",10
801:                dc.b "*****",10
802:                dc.b "*****",10
803:                dc.b "*****",10
804:                dc.b "*****",10
805:                dc.b "*****",10
806:                dc.b "*****",10
807:                dc.b "*****",10
808:                dc.b "*****",10
809:                dc.b "*****",10
810:                dc.b "*****",10
811:                dc.b "*****",10
812:                dc.b "*****",10
813:                dc.b "*****",10
814:                dc.b "*****",10
815:                dc.b "*****",10
816:                dc.b "*****",10
817:                dc.b "*****",10
818:                dc.b "*****",10
819:                dc.b "*****",10
820:                dc.b "*****",10
821:                dc.b "*****",10
822:                dc.b "*****",10
823:                dc.b "*****",10
824:                dc.b "*****",10
825:                dc.b "*****",10
826:                dc.b "*****",10
827:                dc.b "*****",10
828:                dc.b "*****",10
829:                dc.b "*****",10
830:                dc.b "*****",10
831:                dc.b "*****",10
832:                dc.b "*****",10
833:                dc.b "*****",10
834:                dc.b "*****",10
835:                dc.b "*****",10
836:                dc.b "*****",10
837:                dc.b "*****",10
838:                dc.b "*****",10
839:                dc.b "*****",10
840:                dc.b "*****",10
841:                dc.b "*****",10
842:                dc.b "*****",10
843:                dc.b "*****",10
844:                dc.b "*****",10
845:                dc.b "*****",10
846:                dc.b "*****",10
847:                dc.b "*****",10
848:                dc.b "*****",10
849:                dc.b "*****",10
850:                dc.b "*****",10
851:                dc.b "*****",10
852:                dc.b "*****",10
853:                dc.b "*****",10
854:                dc.b "*****",10
855:                dc.b "*****",10
856:                dc.b "*****",10
857:                dc.b "*****",10
858:                dc.b "*****",10
859:                dc.b "*****",10
860:                dc.b "*****",10
861:                dc.b "*****",10
862:                dc.b "*****",10
863:                dc.b "*****",10
864:                dc.b "*****",10
865:                dc.b "*****",10
866:                dc.b "*****",10
867:                dc.b "*****",10
868:                dc.b "*****",10
869:                dc.b "*****",10
870:                dc.b "*****",10
871:                dc.b "*****",10
872:                dc.b "*****",10
873:                dc.b "*****",10
874:                dc.b "*****",10
875:                dc.b "*****",10
876:                dc.b "*****",10
877:                dc.b "*****",10
878:                dc.b "*****",10
879:                dc.b "*****",10
880:                dc.b "*****",10
881:                dc.b "*****",10
882:                dc.b "*****",10
883:                dc.b "*****",10
884:                dc.b "*****",10
885:                dc.b "*****",10
886:                dc.b "*****",10
887:                dc.b "*****",10
888:                dc.b "*****",10
889:                dc.b "*****",10
890:                dc.b "*****",10
891:                dc.b "*****",10
892:                dc.b "*****",10
893:                dc.b "*****",10
894:                dc.b "*****",10
895:                dc.b "*****",10
896:                dc.b "*****",10
897:                dc.b "*****",10
898:                dc.b "*****",10
899:                dc.b "*****",10
900:                dc.b "*****",10
901:                dc.b "*****",10
902:                dc.b "*****",10
903:                dc.b "*****",10
904:                dc.b "*****",10
905:                dc.b "*****",10
906:                dc.b "*****",10
907:                dc.b "*****",10
908:                dc.b "*****",10
909:                dc.b "*****",10
910:                dc.b "*****",10
911:                dc.b "*****",10
912:                dc.b "*****",10
913:                dc.b "*****",10
914:                dc.b "*****",10
915:                dc.b "*****",10
916:                dc.b "*****",10
917:                dc.b "*****",10
918:                dc.b "*****",10
919:                dc.b "*****",10
920:                dc.b "*****",10
921:                dc.b "*****",10
922:                dc.b "*****",10
923:                dc.b "*****",10
924:                dc.b "*****",10
925:                dc.b "*****",10
926:                dc.b "*****",10
927:                dc.b "*****",10
928:                dc.b "*****",10
929:                dc.b "*****",10
930:                dc.b "*****",10
931:                dc.b "*****",10
932:                dc.b "*****",10
933:                dc.b "*****",10
934:                dc.b "*****",10
935:                dc.b "*****",10
936:                dc.b "*****",10
937:                dc.b "*****",10
938:                dc.b "*****",10
939:                dc.b "*****",10
940:                dc.b "*****",10
941:                dc.b "*****",10
942:                dc.b "*****",10
943:                dc.b "*****",10
944:                dc.b "*****",10
945:                dc.b "*****",10
946:                dc.b "*****",10
947:                dc.b "*****",10
948:                dc.b "*****",10
949:                dc.b "*****",10
950:                dc.b "*****",10
951:                dc.b "*****",10
952:                dc.b "*****",10
953:                dc.b "*****",10
954:                dc.b "*****",10
955:                dc.b "*****",10
956:                dc.b "*****",10
957:                dc.b "*****",10
958:                dc.b "*****",10
959:                dc.b "*****",10
960:                dc.b "*****",10
961:                dc.b "*****",10
962:                dc.b "*****",10
963:                dc.b "*****",10
964:                dc.b "*****",10
965:                dc.b "*****",10
966:                dc.b "*****",10
967:                dc.b "*****",10
968:                dc.b "*****",10
969:                dc.b "*****",10
970:                dc.b "*****",10
971:                dc.b "*****",10
972:                dc.b "*****",10
973:                dc.b "*****",10
974:                dc.b "*****",10
975:                dc.b "*****",10
976:                dc.b "*****",10
977:                dc.b "*****",10
978:                dc.b "*****",10
979:                dc.b "*****",10
980:                dc.b "*****",10
981:                dc.b "*****",10
982:                dc.b "*****",10
983:                dc.b "*****",10
984:                dc.b "*****",10
985:                dc.b "*****",10
986:                dc.b "*****",10
987:                dc.b "*****",10
988:                dc.b "*****",10
989:                dc.b "*****",10
990:                dc.b "*****",10
991:                dc.b "*****",10
992:                dc.b "*****",10
993:                dc.b "*****",10
994:                dc.b "*****",10
995:                dc.b "*****",10
996:                dc.b "*****",10
997:                dc.b "*****",10
998:                dc.b "*****",10
999:                dc.b "*****",10
1000:               dc.b "*****",10

```

Listing: LinkList



## CPS AT



### CPS AT I

(siehe Test DOS International 1/89)

80286-12 CPU 8/12 MHz, 1 MB RAM,  
bis 4 MB on board, 0-Wait-State,  
Sockel für Co-Prozessor, EMS, Phoenix Bios,  
64 K ROM mit Setup-Funktion, Uhr u. Kalender,  
1 ser./2 par./Game-Port, Herc. komp. Grafikkarte,  
20 MB Festplatte 38 ms Zugriffszeit,  
1 Lw 5,25" 1,2 MB und 1 Lw 3,5" 720 KB/1,44 MB  
Cherry Tastatur mit sep. Cursorblock,  
14" Flat Screen Monitor s/w oder amber,  
MS-DOS 4.01 dt. und GW-Basic ..... **2998,-**

### CPS AT

CPU 80286-12, 0-Wait-State, 12 MHz,  
640 KB RAM erw. bis 4 MB on board,  
1 Laufwerk 5,25", 1,2 MB TEAC,  
HDD / FDD Contr.,  
20 MB Festplatte 38 ms,  
1 ser./1 par./Game-Port,  
Herc. komp. Grafikkarte,  
Cherry Tastatur MF-II, 102 Tasten,  
14" Monitor s/w,  
MS-DOS 4.01 dt. und GW-Basic ..... **2430,-**

### CPS 386 SX

(siehe Testbericht DOS-TEST 1/90)

80386 SX CPU 16 MHz, 0-Wait-State,  
2 MB RAM, bis 4 MB on board, Sockel für 80387,  
VGA-Grafikkarte, 40 MB Festplatte 28 ms  
1 ser./1 par./Gameport,  
1 Lw 5,25" 1,2 MB und 1 Lw 3,5" 720 KB/1,44 MB  
Cherry Tastatur mit sep. Cursorblock,  
14" VGA Color Monitor  
MS-DOS 4.01 dt. und GW-Basic ..... **4955,-**

### CPS AT III / Cache

80386-25 MHz CPU, 20/25 MHz, 0-Wait  
1 MB RAM, erw. bis 16 MB on board,  
32 KB Cache, Memory Expans. Card  
Sockel für Co-CPU 80387, EMS-fähig,  
2 par./2 ser. Schnittst./Gameport,  
1 Lw 5,25" 1,2 MB und 1 Lw 3,5" 1,44 MB,  
VGA Grafik EIZO MD-B 10,  
SCSI Combi Controller  
157 MB SCSI Festpl. 14 ms  
14" EIZO 9070s Monitor  
MS-DOS 4.01 und GW-Basic  
UNIX V/386 OS ..... **15.500,-**

**Technische Änderungen vorbehalten.**  
Gerätekonfigurationen nach Ihren Wünschen veränderbar.

## CPS Filibabre

**CPS Computertechnik GmbH**  
Braunhirschstraße 29, 3100 Celle  
Tel. 0 51 41 / 3 20 04, Fax 0 51 41 / 38 14 22

**CPS Computertechnik GmbH**  
Großbeerenstraße 5 · D-1000 Berlin 42  
Telefon 0 30 / 706 94 18

## CPS AT

80286-12 CPU 8/12 MHz, 640 KB RAM bis  
4 MB on board, 0-Wait, 1 ser. 2 par. Game-Port,  
Herc. komp. Grafikk., 1 Laufw. TEAC 1,2 MB 5,25",  
Cherry MF II Tastatur ..... **1880,-**  
Gerätekonfiguration nach Ihren Wünschen veränderbar.  
Speichererweiterungen bitte Tagespreis anfragen.

## COMMODORE

AMIGA 500 ..... **995,-**  
AMIGA 500 + 1084 ..... **1525,-**  
AMIGA 2000 ..... **1985,-**  
AMIGA 2000 + 20 MB Harddisk Autoboot  
inkl. Contr. 2090 A ..... **2990,-**  
AMIGA 2000 + Monitor 1084 ..... **2580,-**  
68020 Prozessorkarte ..... Preis a. Anfr.  
68030 Prozessorkarte ..... Preis a. Anfr.

## ERWEITERUNGEN

20 MB Harddisk A 2000  
inkl. 2090 A Contr. .... **1220,-**  
2 MB Erw. int. (A 2000) orig. CBM ..... **1180,-**  
PC/XT Karte inkl. 5,25" LW  
+ MS-DOS + GW-Basic ..... **899,-**  
PC/AT Karte inkl. 5,25" LW  
+ MS-DOS + GW-Basic ..... **2400,-**  
512 KB Erw. A 500 ..... **330,-**

## AMIGA-ZUBEHÖR

LW ext. 3,5" ohne Display ..... **290,-**  
LW ext. 5,25" ohne Display ..... **419,-**  
LW intern 3,5" inkl. Einbausatz ..... **220,-**  
Commodore 1084 ..... **630,-**  
Philips RGB Color CM 8833 ..... **630,-**  
EGA Standard ..... **815,-**



## DRUCKER

OKI Microline 390 Centr. .... Auf Anfr.  
OKI Microline 380 ..... **1138,-**  
OKI QL 400 + 800 Laserline ..... Sonderpreis auf Anfr.  
NEC P 6 Plus Centr. .... **1499,-**  
NEC Silentwriter LC 890,  
Postscript, 3 MB RAM ..... **6995,-**  
Star LC 10 ..... **448,-**  
Star LC 10 II ..... **528.50**  
Star LC 24-10 ..... **680,-**  
Star XB 24-10 ..... **1437,-**  
EPSON LX 400 ..... **448,-**  
EPSON LQ 400 ..... **719,-**  
EPSON LQ 550 ..... **982,-**  
Centronics Drucker kabel ..... **18,80**

Weitere Drucker auf Anfrage

Wir liefern nur mit dt. Handbuch, Seriennummer und  
Herstellergarantie!!! Drucker-Graupimporte mit engl. Hand-  
buch, ohne Seriennummer, ohne Herstellergarantie sind  
bei uns ausgeschlossen.

## MONITORE



EGA Standard ..... **815,-**  
EIZO 9060s ..... **1855,-**  
NEC MULTISYNC 2 A ..... **1550,-**  
TTL 14" Flat Screen mit Fuß sw/amber **235,-**  
Mitsubishi 1481 ..... **1398,-**  
weitere Monitore auf Anfrage

## DISKETTEN



NN 2DD 3,5" 10 Stück ..... **15,00**  
NN 2DD 5,25" 10 Stück ..... **6,90**  
Magix MF 2DD 3,5" ..... **24,50**  
Select MF 2DD 3,5" ..... **33,50**  
Select MD 2DD 5,25" ..... **19,00**  
Fuji MD 2DD 5,25" ..... **27,00**

## FARBEBÄNDER



STAR NL/NG/ND/NR-10, Stück ..... **8,60**  
EPSON LX-800/LQ-500, Stück ..... **7,50**  
PANASONIC KX-P, Stück ..... **9,25**  
OKI ML 320 ..... **7,20**  
OKI ML 390 ..... **10,40**  
NEC 2200 ..... **10,25**  
NEC P6+/P7+ ..... **10,90**  
Star LC 10 ..... **7,00**

## SOFTWARE

ALDUS Pagemaker 3.0 ..... **2150,-**  
GEM Arline 1.0 ..... **1390,-**  
GEM Desktop 1.1 ..... **1200,-**  
MS-DOS Lernprogramm 2.0 ..... **168,-**  
MS-Windows 286 2.1 ..... **480,-**  
MS-Windows 386 2.1 ..... **750,-**  
weitere Software auf Anfrage

**Sämtliche Angebote freibleibend, Zwischenverkauf  
vorbehalten. Wir liefern an Nicht-Kaufleute nur per  
UPS-Nachnahme. Ins Ausland nur per Vorkasse.**

**Fordern Sie unseren Gesamtkatalog gegen  
3,- DM in Briefmarken.**

Es gelten unsere allgemeinen Geschäftsbedingungen.

**Versand nur über Braunschweig!**

× 24 Monate Garantie auf CPS-Peripherie  
× 10 Tage Rückgaberecht auf Hardware  
× Eigener Reparatur-Service

**Nutzen Sie unseren Bequem-Kauf-Kredit!**

Die von uns angegebenen Taktfre-  
quenzen (MHz) beziehen sich  
**nicht** auf den Speedtest nach  
Landmark, sondern sind  
**tatsächliche** Taktfrequenzen!

**CPS**  
weil Preis und  
Leistung stimmen!

Garry Glendown

## Dem CLI Beine machen

### Im Blickpunkt: die ARP-Library

Bei der Programmierung des Amiga gibt es kleine, mittlere und große Schwierigkeiten, die man je nach persönlichem Kenntnisstand besser oder schlechter bewältigen kann. Heraus kommt dann die x-te Speicherverwaltungsroutine oder die 200ste Pattern-Match-Routine. Es ist schon schlimm, daß der Amiga das nicht standardmäßig zur Verfügung stellt...

**D**och ganz so schlimm ist es nun ja auch nicht, denn gemessen an dem Grad der Verbreitung gibt es einen 'Quasi-Standard', der mehr und mehr von den Programmierern genutzt wird: die ARP-Library.

Doch was ist dieses ARP eigentlich? Die ARP-Distribution besteht neben der eigentlichen Library aus der kompletten Dokumentation zu den Routinen, einem Satz DOS-Befehle, die praktisch durchgängig besser als die von Commodore gelieferten sind (natürlich auch inklusive Dokumentation), sowie den Routinen zur Einbindung in verschiedene Programmiersprachen. Neben diesen Sprachen ist aber das Einbinden in andere Umgebungen eigentlich kein Problem, so daß auch eine Nutzung von ARP innerhalb von Amiga- oder GFA-BASIC möglich ist.

### Von den Befehlen...

Die ARP-DOS-Befehle zeichnen sich dadurch aus, daß sie zum einen alle kürzer sind als ihre Amiga-DOS-Äquivalente, zum anderen eine bessere Leistung an den Tag legen. Angefangen mit dem "\*" als Wildcard, den die ARP-Befehle zusätzlich besitzen, über die geringere Anfälligkeit bei Sonderfällen (Beispiel: Geht man mit dem Amiga-DOS-"Search" auf ein Binärfile los, so wird man recht viele "Warning: line <x> to long" bekommen, in denen dann natürlich auch nicht gesucht wird. ARP-Search geht da anstandslos durch...) bis hin zu den wesentlich größeren Möglichkeiten (Beispiel: ARP-"Sort" ist nicht nur

schneller, sondern man hat noch die Möglichkeit, die Sortierbreite und eine Sortierung mit Unterscheidung von Groß- und Kleinschreibung zu bestimmen).

Leider enthalten die Befehle der aktuellen Version noch einige Fehler, die jedoch in der Praxis in Anbetracht der besseren Leistung in Kauf genommen werden können. Doch wie haben es die Programmierer der Befehle überhaupt geschafft, die Befehle kürzer zu machen? Statt Routinen wie die Abfrage der Übergabeparameter in jedem Programm abzulegen, wird einfach eine Routine in der ARP-Library verwendet. Werden dann später Erweiterungen oder Fehlerkorrekturen vorgenommen, zum Beispiel beim Pattern-Matching oder ähnlichem, so sind automatisch alle Programme korrigiert, wenn die Library ausgetauscht wird. So laufen natürlich Programme, die mit ARP 1.1 programmiert wurden, auch mit der Version 1.3 einwandfrei. Funktionen wie der Filerequester zeigen nicht mehr den alten, sondern den neuen Requester.

### ...zur Library

Der Datenteil der Library-Struktur, also die eigentliche Librarybase, enthält, im Vergleich zu den Möglichkeiten der ARP-Library, recht wenig Daten. Interessant sind lediglich die Zeiger auf DOS-, Graphics- und Intuition-Library, die man bei erfolgreichem Öffnen der ARP-Library nur noch übernehmen muß. Dagegen ist jedoch die Größe der Sprungtabelle wahrhaft riesig; sie übertrifft mit ihren 105 Routinen sogar die schon

recht große Graphics-Library. In Tabelle 1 befindet sich eine Auflistung aller Routinen mit Offset sowie den Parametern (mit Registerangabe für Assemblerprogrammierer).

Um nun eigene Programme mit diesen Routinen 'anzureichern', wollen wir an dieser Stelle erst einmal systematisch nachsehen, welche Routinen besonders interessant sind. Dazu müssen natürlich Vorbereitungen getroffen werden. Entweder man benutzt die von der "arp.lib" bereitgestellten Routinen (in diesem Fall wird die Library etc. automatisch geöffnet), oder man öffnet die ARP-Library selber. In diesem Fall muß man mit "a.lib" linken.

Doch nun zu den 'wirklichen' Library-Routinen. Als erstes fallen da natürlich die Routine "Printf", "FPrintf" und "SPrintf" ins Auge. Diese sorgen bei normaler Anwendung der Routinen aus der "c.lib" dafür, daß jedes beliebige Programm automatisch um einiges wächst. Zur Verwendung braucht man, dank Argument-Kompatibilität, nur die Namen in den Aufrufen zu ändern oder aber Zeilen wie:

```
#define printf Printf
#define sprintf SPrintf
#define fprintf FPrintf
```

"Im selben Aufwasch" entledigen wir uns noch der Puts-Funktion, die wir durch "Puts" ersetzen. Nicht zu vergessen die Routinen zum String-Vergleich, "Strcmp" und "Strncmp". Also noch drei Defines:

```
#define puts Puts
#define strcmp Strcmp
#define strncmp Strncmp
```

Zum Abschluß wollen wir noch einen Blick auf ein recht heikles Kapitel werfen: die Pattern-Match-Routinen, deren vielzählige Varianten man mit dem Satz "oft kopiert - nur selten erreicht" charakterisieren kann. Sie kennen das Problem: Wie oft war man schon der Meinung, die wirklich optimale Routine gefunden zu haben, und dann trat irgendwann ganz nebenbei ein Fall auf, bei der sie mit Pauken und Trompeten "durch die Prüfung gerasselt" ist... Ganz zu schweigen von einigen Funktionen, bei der nur bestimmte Zeichen zugelassen sind, und vieles mehr. Alles das wird von den ARP-Routinen doch recht perfekt beherrscht. Und da das Ganze noch recht einfach einzusetzen ist, wollen wir uns hier

ein kurzes Code-Fragment anschauen, in dem diese Routinen angewandt werden. Gegeben sei ein String "a", dessen Vorkommen wir in verschiedenen eingelesenen Zeilen testen wollen. Zu diesem Zweck muß er noch vorbereitet werden. Dies geschieht folgendermaßen: Wir kopieren ihn in einen anderen String-Puffer hinein, wobei vorne und hinten noch ein '\*' angefügt wird. Um auch noch unabhängig von Groß- und Kleinschreibung zu sein, wird er vorher noch mittels "StrToUpper" in Großschrift umgewandelt:

```
VOID StrToUpper(txt)
char *txt;
{
    while (*txt) {
        *txt=ToUpper(*txt);
        txt++;
    }
}

int Match(a)
char *a;
{
    char b[256],c[256];

    strcpy(b,"*");
    strcat(b,a);
    strcat(b,"*");
    StrToUpper(b);
```

Nun steht also in "b" der gesuchte Text in Großschrift. Damit wir ihn mit der Pattern-Match-Routine verwenden können, muß er noch bearbeitet werden, was mit "PreParse" gemacht wird. Dabei werden die verschiedenen Darstellungsarten auf ein internes Format gebracht:

```
PreParse(a,c);
```

Der String in "c" kann nun mittels der Routine "PatternMatch" zum Vergleichen verwendet werden. Bei einem Array in "nums" von "g"-Strings kann nun beispielsweise mit folgender Schleife durchsucht werden:

```
for (t=0;t<g;t++) {
    strcpy(b,nums[t]);
    StrToUpper(b);
    if (PatternMatch(c,b))
        Printf("Gefunden: %s\n",
            nums[t]);
}
```

Damit beenden wir den ersten Teil unseres kleinen ARP-Exkurses. In der nächsten Ausgabe der AMIGA DOS gehen wir auf die Programmierung des Filerequesters sowie einiger anderer Routinen ein.

(vb)



## Alphabetische Liste der ARP-Routinen

AddDADevs	\$-204	-516 dalist/A0, Select/D0
AddDANode	\$-1FE	-510 Data/A0, *DALst/A1, Len/D0, ID/D1
AddResidentPrg	\$-246	-582 Segment/D1, Name/A0
ArpAlloc	\$-180	-384 Size/D0
ArpAllocFreq	\$-294	-660 -
ArpAllocMem	\$-186	-390 Size/D0, Reqs/D1
ArpDupLock	\$-192	-402 Lock/D1
ArpExit	\$-17A	-378 ReturnCode/D0, Fault/D2
ArpLock	\$-198	-408 Name/D1, Accessmode/D2
ArpOpen	\$-18C	-396 Name/D1, Accessmode/D2
ArpOpenLibrary	\$-28E	-654 Lib/A1, Version/D0
Assign	\$-150	-336 Name/A0, Physical/A1
ASyncRun	\$-222	-546 Command/A0, Args/A1, PCB/A2
Atol	\$-102	-258 String/A0
BaseName	\$-276	-630 PathName/A0
BtoCStr	\$-162	-354 CStr/A0, BStr/D0, MaxLen/D1
CheckAbort	\$-10E	-270 (*Func)/A1
CheckBreak	\$-114	-276 Mask/D1, (*Func)/A1
CheckSumPrg	\$-26A	-618 Node/D1
Close	\$-024	-36 /* = DOS-Funktion */
CloseWindowSafely	\$-12C	-300 Window/A0, MoreWindows/A1
CompareLock	\$-1C8	-456 Lock1/D0, Lock2/D1
CreateDir	\$-078	-120 /* = DOS-Funktion */
CreatePort	\$-132	-306 Name/A0, Priority/D0(0:8)
CreateProc	\$-08A	-138 /* = DOS-Funktion */
CreateTaskResList	\$-1D4	-468 -
CtoBStr	\$-168	-360 CStr/A0, BStr/D0, MaxLen/D1
CurrentDir	\$-07E	-126 /* = DOS-Funktion */
DateStamp	\$-0C0	-192 /* = DOS-Funktion */
Delay	\$-0C6	-198 /* = DOS-Funktion */
DeleteFile	\$-048	-72 /* = DOS-Funktion */
DeletePort	\$-138	-312 Port/A1
DeviceProc	\$-0AE	-174 /* = DOS-Funktion */
DosAllocMem	\$-156	-342 Size/D0
DosFreeMem	\$-15C	-348 MemBlock/A1
DupLock	\$-060	-96 /* = DOS-Funktion */
EscapeString	\$-108	-264 String/A0
Examine	\$-066	-102 /* = DOS-Funktion */
Execute	\$-0DE	-222 /* = DOS-Funktion */
Exit	\$-090	-144 /* = DOS-Funktion */
ExNext	\$-06C	-108 /* = DOS-Funktion */
FileRequest	\$-126	-294 FileRequester/A0
FindCLI	\$-1A4	-420 TaskNum/D0
FindFirst	\$-1B6	-438 Pattern/D0, AnchorPath/A0
FindNext	\$-1BC	-444 AnchorPath/A0
FindTaskResList	\$-1CE	-462 -
FPrintf	\$-0EA	-234 File/D0, String/A0, *args/A1
FreeAccess	\$-1F2	-498 Tracker/A1
FreeAnchorChain	\$-1C2	-450 AnchorPath/A0
FreeDAList	\$-1F8	-504 daNode/A1
FreeResList	\$-1DA	-474 -
FreeTaskResList	\$-174	-372 -

FreeTrackedItem	\$-1E0	-480 Tracker/A1
GADS	\$-0FC	-252 line/A0, len/D0, help/A1, args/a2, tplate/A3
GetAccess	\$-1EC	-492 Tracker/A1
GetDevInfo	\$-16E	-366 DevInfoNode/A2
Getenv	\$-11A	-282 String/A0, Buffer/A1, Size/D0
GetKeywordIndex	\$-288	-648 KWrd/A0, template/A1
GetTracker	\$-1E6	-486 -
Info	\$-072	-114 /* = DOS-Funktion */
InitStdPacket	\$-144	-324 Action/D0, Args/A0, Packet/A1, ReplyPort/A2
Input	\$-036	-54 /* = DOS-Funktion */
IoErr	\$-084	-132 /* = DOS-Funktion */
IsInteractive	\$-0D8	-216 /* = DOS-Funktion */
LDiv	\$-25E	-606 Dividend/D0, Divisor/D1
LMod	\$-264	-612 Dividend/D0, Divisor/D1
LMult	\$-258	-600 Num1/D0, Num2/D1
LoadPrg	\$-228	-552 Name/D1
LoadSeg	\$-096	-150 /* = DOS-Funktion */
Lock	\$-054	-84 /* = DOS-Funktion */
ObtainResidentPrg	\$-240	-576 Name/A0
Open	\$-01E	-30 /* = DOS-Funktion */
Output	\$-03C	-60 /* = DOS-Funktion */
ParentDir	\$-0D2	-210 /* = DOS-Funktion */
PathName	\$-14A	-330 Lock/D0, Dest/A0, NumberNames/D1
PatternMatch	\$-1B0	-432 Pattern/A0, String/A1
PreParse	\$-22E	-558 Source/A0, Dest/A1
Printf	\$-0E4	-228 String/A0, *args/A1
Puts	\$-0F0	-240 String/A1
QSort	\$-1AA	-426 Ptr/A0, RegSize/D0, ByteSize/D1, UserFunc/A1
Read	\$-02A	-42 /* = DOS-Funktion */
ReadLine	\$-0F6	-246 Buffer/A0
ReleaseResidentPrg	\$-27C	-636 Segment/D1
RemResidentPrg	\$-24C	-588 Name/A0
Rename	\$-04E	-78 /* = DOS-Funktion */
RListAlloc	\$-19E	-414 ResList/A0, Size/D0
Seek	\$-042	-66 /* = DOS-Funktion */
SendPacket	\$-13E	-318 Action/D0, Args/A0, Handler/A1
SetComment	\$-0B4	-180 /* = DOS-Funktion */
Setenv	\$-120	-288 String/A0, Buffer/A1
SetProtection	\$-0BA	-186 /* = DOS-Funktion */
SpawnShell	\$-222	-546
SPrintf	\$-282	-642 buffer/D0, String/A0, *args/A1
StampToStr	\$-234	-564 DateTime/A0
Strcmp	\$-20A	-522 S1/A0, S2/A1
Strncmp	\$-210	-528 S1/A0, S2/A1, Len/D0
StrtoStamp	\$-23A	-570 DateTime/A0
SyncRun	\$-21C	-540 FileName/A0, Args/A1, Input/D0, Output/D1
TackOn	\$-270	-624 Path/A0, File/A1
Toupper	\$-216	-534 Char/D0
UnLoadPrg	\$-252	-594 Segment/D1
UnLoadSeg	\$-09C	-156 /* = DOS-Funktion */
UnLock	\$-05A	-90 /* = DOS-Funktion */
WaitForChar	\$-0CC	-204 /* = DOS-Funktion */
Write	\$-030	-48 /* = DOS-Funktion */

## Lister ARP-Befehle, nach Offset sortiert

Open	\$-01E	-30 /* = DOS-Funktion */
Close	\$-024	-36 /* = DOS-Funktion */
Read	\$-02A	-42 /* = DOS-Funktion */
Write	\$-030	-48 /* = DOS-Funktion */
Input	\$-036	-54 /* = DOS-Funktion */
Output	\$-03C	-60 /* = DOS-Funktion */
Seek	\$-042	-66 /* = DOS-Funktion */
DeleteFile	\$-048	-72 /* = DOS-Funktion */
Rename	\$-04E	-78 /* = DOS-Funktion */
Lock	\$-054	-84 /* = DOS-Funktion */
UnLock	\$-05A	-90 /* = DOS-Funktion */
DupLock	\$-060	-96 /* = DOS-Funktion */
Examine	\$-066	-102 /* = DOS-Funktion */
ExNext	\$-06C	-108 /* = DOS-Funktion */
Info	\$-072	-114 /* = DOS-Funktion */
CreateDir	\$-078	-120 /* = DOS-Funktion */
CurrentDir	\$-07E	-126 /* = DOS-Funktion */
IoErr	\$-084	-132 /* = DOS-Funktion */
CreateProc	\$-08A	-138 /* = DOS-Funktion */
Exit	\$-090	-144 /* = DOS-Funktion */
LoadSeg	\$-096	-150 /* = DOS-Funktion */
UnLoadSeg	\$-09C	-156 /* = DOS-Funktion */
DeviceProc	\$-0AE	-174 /* = DOS-Funktion */
SetComment	\$-0B4	-180 /* = DOS-Funktion */
SetProtection	\$-0BA	-186 /* = DOS-Funktion */
DateStamp	\$-0C0	-192 /* = DOS-Funktion */
Delay	\$-0C6	-198 /* = DOS-Funktion */
WaitForChar	\$-0CC	-204 /* = DOS-Funktion */
ParentDir	\$-0D2	-210 /* = DOS-Funktion */
IsInteractive	\$-0D8	-216 /* = DOS-Funktion */
Execute	\$-0DE	-222 /* = DOS-Funktion */
Printf	\$-0E4	-228 String/A0, *args/A1
FPrintf	\$-0EA	-234 File/D0, String/A0, *args/A1
Puts	\$-0F0	-240 String/A1
ReadLine	\$-0F6	-246 Buffer/A0
GADS	\$-0FC	-252 line/A0, len/D0, help/A1, args/a2, tplate/A3
Atol	\$-102	-258 String/A0
EscapeString	\$-108	-264 String/A0
CheckAbort	\$-10E	-270 (*Func)/A1
CheckBreak	\$-114	-276 Mask/D1, (*Func)/A1
Getenv	\$-11A	-282 String/A0, Buffer/A1, Size/D0
Setenv	\$-120	-288 String/A0, Buffer/A1
FileRequest	\$-126	-294 FileRequester/A0
CloseWindowSafely	\$-12C	-300 Window/A0, MoreWindows/A1
CreatePort	\$-132	-306 Name/A0, Priority/D0(0:8)
DeletePort	\$-138	-312 Port/A1
SendPacket	\$-13E	-318 Action/D0, Args/A0, Handler/A1
InitStdPacket	\$-144	-324 Action/D0, Args/A0, Packet/A1, ReplyPort/A2
PathName	\$-14A	-330 Lock/D0, Dest/A0, NumberNames/D1
Assign	\$-150	-336 Name/A0, Physical/A1
DosAllocMem	\$-156	-342 Size/D0
DosFreeMem	\$-15C	-348 MemBlock/A1
BtoCStr	\$-162	-354 CStr/A0, BStr/D0, MaxLen/D1

CtoBStr	\$-168	-360 CStr/A0, BStr/D0, MaxLen/D1
GetDevInfo	\$-16E	-366 DevInfoNode/A2
FreeTaskResList	\$-174	-372 -
ArpExit	\$-17A	-378 ReturnCode/D0, Fault/D2
ArpAlloc	\$-180	-384 Size/D0
ArpAllocMem	\$-186	-390 Size/D0, Reqs/D1
ArpOpen	\$-18C	-396 Name/D1, Accessmode/D2
ArpDupLock	\$-192	-402 Lock/D1
ArpLock	\$-198	-408 Name/D1, Accessmode/D2
RListAlloc	\$-19E	-414 ResList/A0, Size/D0
FindCLI	\$-1A4	-420 TaskNum/D0
QSort	\$-1AA	-426 Ptr/A0, RegSize/D0, ByteSize/D1, UserFunc/A1
PatternMatch	\$-1B0	-432 Pattern/A0, String/A1
FindFirst	\$-1B6	-438 Pattern/D0, AnchorPath/A0
FindNext	\$-1BC	-444 AnchorPath/A0
FreeAnchorChain	\$-1C2	-450 AnchorPath/A0
CompareLock	\$-1C8	-456 Lock1/D0, Lock2/D1
FindTaskResList	\$-1CE	-462 -
CreateTaskResList	\$-1D4	-468 -
FreeResList	\$-1DA	-474 -
FreeTrackedItem	\$-1E0	-480 Tracker/A1
GetTracker	\$-1E6	-486 -
GetAccess	\$-1EC	-492 Tracker/A1
FreeAccess	\$-1F2	-498 Tracker/A1
FreeDAList	\$-1F8	-504 daNode/A1
AddDANode	\$-1FE	-510 Data/A0, *DALst/A1, Len/D0, ID/D1
AddDADevs	\$-204	-516 dalist/A0, Select/D0
Strcmp	\$-20A	-522 S1/A0, S2/A1
Strncmp	\$-210	-528 S1/A0, S2/A1, Len/D0
Toupper	\$-216	-534 Char/D0
SyncRun	\$-21C	-540 FileName/A0, Args/A1, Input/D0, Output/D1
SpawnShell	\$-222	-546
ASyncRun	\$-222	-546 Command/A0, Args/A1, PCB/A2
LoadPrg	\$-228	-552 Name/D1
PreParse	\$-22E	-558 Source/A0, Dest/A1
StampToStr	\$-234	-564 DateTime/A0
StrToStamp	\$-23A	-570 DateTime/A0
ObtainResidentPrg	\$-240	-576 Name/A0
AddResidentPrg	\$-246	-582 Segment/D1, Name/A0
RemResidentPrg	\$-24C	-588 Name/A0
UnLoadPrg	\$-252	-594 Segment/D1
LMult	\$-258	-600 Num1/D0, Num2/D1
LDiv	\$-25E	-606 Dividend/D0, Divisor/D1
LMod	\$-264	-612 Dividend/D0, Divisor/D1
ChecksumPrg	\$-26A	-618 Node/D1
TackOn	\$-270	-624 Path/A0, File/A1
BaseName	\$-276	-630 PathName/A0
ReleaseResidentPrg	\$-27C	-636 Segment/D1
SPrintf	\$-282	-642 buffer/D0, String/A0, *args/A1
GetKeywordIndex	\$-288	-648 KWrd/A0, template/A1
ArpOpenLibrary	\$-28E	-654 Lib/A1, Version/D0
ArpAllocFreq	\$-294	-660 -



# +++ VESALIA TOP ANGEBOTE +++

## Winner Autoboot-Filecard und Autoboot-Harddisk

Kapazität 3,5" HD	Zugriffszeit + Lesegeschw.	Filecard + AB-Modul	Autoboot- Filecard	A 500/ A 1000
21 MB =	40 mS/ca. 410 ■■■	798,-	898,-	998,-
31 MB =	40 mS/ca. 445 ■■■	898,-	998,-	1098,-
47 MB +	28 mS/ca. 470 KB	1098,-	1198,-	1298,-
63 MB +	28 mS/ca. 475 KB	1298,-	1398,-	1498,-
133 MB +	15 mS/ca. 500 ■■■	2468,-	2568,-	2668,-

= mit Parkprogramm, + Autopark

## Sonderaktion Ausgabe 5.90

31 MB Filecard  
autoboot ab 1.2  
16 MB und 15 MB  
Millionenkart.

**798,-**

Harddisk und Filecard bereits formatiert und installiert mit ■■■ 1.3. Autoboot sofort nach dem Einschalten mit FFSsystem auch unter Kick. 1.2. Mit Installations-Software und Beschreibung. HD-Gehäuse für A 500/1000 als Monitoruntersatz (33 x 33 = ■ cm) durch den Einbau von 3,5" HD auch für Amiga 500/1000, sind diese Festplatten besonders leise, bei gleichzeitig geringerer Wärmeentwicklung. 12 Monate Garantie.

## Winner Ramkarte 179,-

512 KB abschaltbar, mit akkugepufferter Uhr.  
Megabittechnologie

## 2 MB-Box A 500 698,-

Mit durchgeführtem Bus autokonfigurierend  
und abschaltbar.

## 8 MB-Karte A 2000 848,-

Mit 2 MB bestückt.

## 3,5" Winner-Drive 219,-

Externes Amiga-Laufwerk, abschaltbar, nur 1" hoch mit durchgeführtem Bus.

## 3,5" A 2000 intern 155,-

Wie org. Commodore, komplett mit Zubehör.

## Amiga-Bremse 39,50

Highscore Killer für schnelle Games und  
Bildschirmfotos.

## Bit Fat Agnus 159,-

Mit Einbauanleitung für Amiga 500 und 2000 B  
jetzt 1 MB CHIP-RAM-MEM

## Umschaltplatine 59,-

für 1 org. Kick-Rom und 2 EPROM-Versionen

## Festplattengehäuse 68,-

für Amiga 500/1000 mit allen Bohrungen

## No Name Disketten 2 DD

3,5" 10 Stück	12,90
3,5" 50 Stück	62,-
3,5" 100 Stück	115,-

## Y-C Genlock 1120,-

RGB-Bandbreite 10 MHz. S-VHS Anschluß.  
Wandler von RGB nach Pal usw.

## De Luxe View 4.1 378,-

Der Testsieger

## Fast Lightning 29,-

Das ideale Kopierprogramm für 2-3 Laufwerke. Vier Kopiermodi auch für Atari.

## 512 KB-Karte 149,-

Abschaltbar, mit akkugepufferter Uhr.  
16 ■ 41256 gesockelt.

## 2 MB-Box A-1000 698,-

Mit durchgeführtem Bus autokonfigurierend  
und abschaltbar.

## 8 MB-Karte A 2000 1298,-

Mit 4 MB bestückt.

## 3,5" Amiga extern 189,-

Abschaltbar, unser Renner. Mit durchgeführtem Bus.

## 3,5" A 500 intern 175,-

als DF0 Ersatzlaufwerk.

## Joy-Maus-Adapter 45,-

für gleichzeitigen Anschluß von Maus und  
Joystick, mit LED.

## ROM-ROM-Umschaltmodul

Mit Kick-ROM 1.3	99,-
Mit Kick-ROM 1.2	99,-
Modul o. Kick-ROM	39,-

## Filcardblech 19,-

für 3,5" Harddisk und OMTI-Controller

## Schaltnetzteil 119,-

50 Watt, 12 Volt plus und minus 5 Volt plus

## Winner-Midi A 500 89,-

## Winner-Midi A 1000 89,-

## Sounddigitizer 500 89,-

## Sounddigitizer 2000 89,-

## Pal-Genlock V 1.3 548,-

Überblenden von einer Bildquelle zur anderen.  
Integrierter Splitter.

## Digi View Gold 298,-

Für Amiga 500/2000

## Turbo Copy 2.0 19,-

Für zwei Laufwerke sehr sicher und schnell.  
Sollte jeder besitzen.

## 2 MB-Karte 598,-

Abschaltbar, mit akkugepufferter Uhr,  
16 ■ 511000 gesockelt.

## 4 MB-Box A 1000 1398,-

Mit durchgeführtem Bus autokonfig. auf 2/4  
MB schalt- und abschaltbar.

## 8 MB-Karte A 2000 1998,-

Komplett bestückt.

## 5,25" Winner Drive 269,-

Extern für alle Amigas 40/80 Track und abschaltbar mit durchgeführtem Bus.

## 5,25" Winner Drive 269,-

für A 2000 intern, komplett mit electr. Boot-selector.

## PowerFire 29,-

Dauerfeuerinterface für Joystick und Maus.  
Für jedes Game optimal einstellbar.

## Boot-Strap 368,-

Umschaltbar von Kick-ROM auf Disketten-Version wie beim Amiga 1000.

## OMTI-Adapterplatine 50,-

komplett bestückt, adr. für ALF-Software u. a.

## OMTI 5520 135,-

## OMTI 5528 149,-

## Kabelsatz 10,-

## Citizen Swift 24 998,-

Der Testsieger unter den 24 Nadel-Druckern.  
Mit Drucker- und dt. Handbuch.

## Digi-Splitt Jr. 428,-

Der Testsieger. Vollautomatischer RGB-Splitter. Für z. B. Digi View 4.0.

## DLS V 2.8 für A 500 225,-

Mit neuer Software

## BootBlockGenerator 19,-

Zum Erstellen eines eigenen Vorspanns in den  
Bootblock, auch mit Sound.

## Vesalia Computer

Magdalenenweg 4 · 4230 Wesel · Telefon 0281/65466 · Telefax 0281/64066 · Montags – Freitags 9–17 Uhr (nur Versand)  
Industriestraße 25 · 4236 Hamminkeln · Telefon 02852/1069 (ab 2.5.90) Mo. – Fr. 9–17 Uhr (Produktion und Ausstellung)

**F**ile-Monitore haben ein großes Anwendungsfeld. Man kann sie zum Eindeutschen, zum Patchen oder einfach mal zum Anschauen von Objektdateien verwenden. Auf dem Amiga sind gute File-Monitore jedoch Mangelware. Die meisten belegen über 30 kByte auf Diskette, da sie in C geschrieben sind. Die Geschwindigkeit läßt demnach natürlich auch zu wünschen übrig.

## FileMonitor V2.0 – schnell und komfortabel

Der FileMonitor V2.0, der komplett in Assembler programmiert wurde, behebt die Mängel und bietet zudem eine komfortable Benutzeroberfläche. Diese wird von einer Vielfalt an Funktionen begleitet. Nach dem Eintippen können Sie das Programm durch die Eingabe seines Dateinamens im CLI starten. Optional können Sie die zu ladende Datei angeben, also zum Beispiel "FileMonitor C:\NewCLI". Ein Start von der Workbench ist ebenfalls möglich, wenn Sie vorher eine passende '.info'-Datei erstellt haben.

Haben Sie keinen Parameter angegeben, so erscheint ein Requester, der Sie auffordert, die zu editierende Datei anzugeben. Um hierbei die größtmögliche Benutzerfreundlichkeit zu erreichen, ohne daß das Programm zu lang wird, wurde zur Dateiauswahl die Dateiauswahlbox aus der 'arp.library' verwendet. ARP steht für "Amiga-DOS Replacement Project" und wurde von einer eifrigen Programmiererguppe geschaffen, um die CLI-Befehle durch kürzere Assemblerprogramme zu ersetzen. Da ARP Public Domain ist, besitzt es eine hohe Verbreitung und ist leicht zu besorgen. Es lohnt sich wirklich (LIST-Befehl mit 2500 Bytes? – Im ARP schon realisiert!). Aber auch wenn Sie die 'arp.library' nicht in Ihrem LIBS-Verzeichnis haben, können Sie FileMonitor verwenden. Dann können Sie den Dateinamen über einen einfacheren Requester eingeben. Ein eingegebener Leerstring bedeutet Abbruch (falls Sie die Maus nicht zur Hand haben, um "Abbruch" anzuklicken). Wenn alles glatt gegangen ist,

Stefan Hochmuth

## Wie sieht's denn da aus?

### FileMonitor – Einblick in jedes Programm

**Fast jeder zeigt heute Einsicht: Die Grenzen gehen auf, die Abrüstung findet statt, der Umweltschutz hält endlich Einzug – nur manche Amiga-Programme weigern sich immer noch hartnäckig, Einsicht in sich selbst zu geben. Dabei gibt es genug Anlässe, die Bytes näher zu besehen und, falls notwendig, zu ändern, ■■■■ Beispiel bei fehlerhaften Disketten, die sich im DOS mit Read- oder Write-Error bemerkbar machen, oder bei Programmen, die mal eben auf die schnelle korrigiert werden müssen, ohne den Source-Code zum wiederholten Male anzeigen zu lassen. Wie man solche Momente gut übersteht? Ganz einfach – mit FileMonitor.**

wird jetzt die Datei geladen und angezeigt.

**Achtung!** Machen Sie sich auf jeden Fall vorher eine Sicherheitskopie von der zu editierenden Datei!

### Bewegung, Bewegung!

Als erstes ist es natürlich wichtig zu wissen, wie man den Cursor bewegt. Um den Cursor auf eine Stelle zu bewegen, die bereits im Display sichtbar ist, genügt ein einfacher Mausklick. Übrigens gibt es natürlich nicht nur einen Cursor, sondern deren zwei, die immer gleichzeitig bewegt werden (einer im Hex-Zahlenfeld, der andere im zu-

gehörigen ASCII-Feld). Sehen Sie oben und unten im File-Monitor-Fenster die beiden orangeroten (Farbe 3), beschrifteten Leisten? Auf ihnen sind insgesamt sechs Funktionen dem Beweg innerhalb der Datei gewidmet. Angewählt werden diese durch einfaches Anklicken des zugehörigen Textes. Die Funktion wird dann so lange ausgeführt, wie man die linke Maustaste gedrückt hält. Damit kann man die Datei bequem mausgesteuert durchblättern (Seite vorwärts/rückwärts) oder durchscrollen (Zeile vorwärts/ rückwärts) bzw. den Cursor auf den Dateianfang oder das Dateiende setzen. All diese Funktionen sind auch über Tastatur er-

reichbar. Die Cursortasten sind wie in vielen Editoren belegt:

**Cursortaste**  
**links/rechts/oben/unten**  
Cursor in entsprechende Richtung

**SHIFT + oben**  
Eine Bildschirmseite rückwärts

**SHIFT + unten**  
Eine Bildschirmseite vorwärts

**ALT + oben/unten**  
Dateianfang/Dateiende

Auch über die Menüsteuerung im Positionsmenü läßt sich einiges erreichen:

**Dateianfang**  
Rechte Amiga-Taste + T

**Dateiende**  
Rechte Amiga-Taste + B

**Hexoffset eingeben**  
Rechte Amiga-Taste + G

Auf die letzte Eingabe hin erscheint ein Requester, in dem Sie die gewünschte Dateiposition als Hexadezimalzahl angeben können. Nach Eingabe von 'Return' springt der Cursor an die gewünschte Stelle. Haben Sie eine 'Nicht-Hex-Zahl' angegeben, so erhalten Sie eine Fehlermeldung. Wenn die Hex-Zahl größer als die Dateilänge (wird übrigens links unten im FileMonitor neben dem Dateizeiger als Hexzahl angezeigt) sein sollte, erfolgt keine Reaktion.

### Viele Wege führen zum Editieren

Der Editiermodus ist gewählt, wenn einer der beiden Cursor aus einem ausgefüllten Rechteck und nicht mehr nur aus einem Rahmen besteht. Die Hex- bzw. die ASCII-Hälfte befindet sich dann im Editiermodus. Um in diesen zu gelangen, gibt es einige Möglichkeiten:

A) Mit der Maus: Machen Sie einen Doppelklick an die Stelle im HEX- bzw. ASCII-Display, an der Sie mit dem Editieren beginnen möchten.

B) Über die Leiste: Klicken Sie 'Ed HEX' an, um die Hex-Hälfte zu editieren, bzw. 'Ed ASCII' für die Texthälfte.

C) Über Menü: Wählen Sie 'HEX-/ASCII-Hälfte editieren' oder 'Rechte Amiga-Taste + X / A' aus.

D) F10-Taste: Editiermodus an/aus



Bild 1. Mit dem FileMonitor auf Byte-Suche



E) TAB-Taste: Editiermodus anschalten und zwischen HEX-/ASCII-Editiermodus wechseln

Auch im Editiermodus besitzen Sie weiterhin alle Möglichkeiten, sich in der Datei zu bewegen. Das Editieren selbst erfolgt im ASCII-Modus durch einfache Eingabe, auf der Hex-Seite wird nur 0 bis 9 und A bis F akzeptiert. Das Ergebnis Ihrer Eingabe wird jedoch erst nach dem Tippen beider Hex-Nibble (= "Hex-Zahlen-Hälften") im Display erscheinen. Im Editiermodus kann man sehr leicht durch ungewollte Tastendrücke ebenso ungewollte Veränderungen vornehmen. Schalten Sie ihn deswegen nach dem Editieren vorsichtshalber sofort wieder aus, indem Sie entweder "Ed aus" mit der Maus in der Menüleiste anklicken oder "Editieren aus" im Editierenmenü anwählen (bzw. rechte Amiga-Taste + "-"). Drücken Sie bei der Short-cut-Wahl gewissenhaft die richtigen Tasten, um dabei nicht versehentlich zu editieren.

F10-Taste: Editiermodus an/aus  
Wenn man beim Editieren einen Fehler macht, gibt es eine Möglichkeit, diesen wieder rückgängig zu machen. Und zwar mit der Seiten-Undo-Funktion aus dem Editierenmenü. Bei Aufruf wird alles Editierte, was in der Zwischenzeit nicht vom Bildschirm verschwunden ist, wieder in den vorigen Zustand gesetzt. Wenn Sie Routinen verwenden, die das ganze Display neu aufbauen, wird dieser interne Undo-Buffer folglich auch neu geladen. Meiden Sie deshalb folgende Funktionen, wenn Sie eventuell ein Undo machen wollen:

- Datei anfang/-ende
- Seite vor-/rückwärts

Programmname	Suchtyp	Suche nach	Ersetze durch	Funktion
AmigaBasic	HEX	323C00C8	323C00FF	volle Fenstergröße
C:NewCLI	ASCII	0/0/640/100	0/0/640/256	" "
C:Setclock	HEX	4EAEFF22	4E714E71	DATE-Befehl nicht ausführen (spart Zeit)
GenAm2	HEX	028000BC	028000F4	angenehme Fenstergröße

Abb. 1. Einige Beispiele, Bytes gezielt zu ändern

- Suchen
- Hexoffset eingeben

Bei normalen Cursorbewegungen (bzw. Zeile vor/zurück) wird der interne Buffer mitgescrollt, das heißt, wenn nach oben gescrollt wird, verschwindet die erste Zeile aus dem Buffer, alle anderen rücken eine Zeile nach und die neu erscheinende wird wieder in den Buffer geschrieben. Die Seiten-Undo-Funktion erweist sich manchmal als sehr nützlich.

## Suchen und Finden

Selbstverständlich besitzt der FileMonitor auch diverse Funktionen zur String- und Hex-Zeichenfolgesuche. Mit den ersten beiden Menüpunkten im Suchen-Menü können Sie bestimmen, ob Sie eine Fallunterscheidung von Groß- und Kleinbuchstaben wünschen. Voreingestellt ist, daß Groß- und Kleinschreibung gleich behandelt wird. Bei "String suchen" erscheint wieder ein Requester, der Sie nach der zu suchenden Zeichenfolge fragt. Nach der Bestätigung mit Return-Taste wird nach dem String gesucht. Klicken Sie "Vorige!" an, um die vorhergehende Zeichenfolge zu suchen. Wird sie nicht gefunden, so bleibt der Cursor an der alten Stelle. Um ohne Neueingabe des Strings vor- oder rückwärts

die gleiche Zeichenfolge suchen zu können, wählen Sie im Menü die entsprechenden Funktionen oder deren Shortcuts an. Gleiches gilt für das Suchen von Hex-Folgen, allerdings müssen Sie bei deren Eingabe beachten, daß immer zwei Zeichen für ein ganzes Byte stehen. Eine Eingabe wie "3AB" wird daher zurückgewiesen.

Bleibt nur noch das Projektmenü zu erklären. Das Laden wurde bereits anfangs beschrieben. Die Bedienung der Speichern-Funktion ist entsprechend. Nochmals die Warnung: Arbeiten Sie nur mit Backups! Um einen Datenverlust zu vermeiden, sind im FileMonitor übrigens folgende Sicherheitsabfragen eingebaut:

### A) Änderungen verlieren?

Wenn Sie eine Datei editiert haben und den FileMonitor verlassen oder eine neue Datei laden wollen, obwohl Sie vorher nicht gespeichert haben, erscheint diese Abfrage.

### B) Alte Datei überschreiben?

Existiert die beim Speichern angegebene Datei bereits, so müssen Sie hier nochmals bestätigen, um wirklich zu speichern.

Wie bei allen Requesterfenstern im FileMonitor, bedeutet auch hier das Drücken einer beliebigen Taste einen Abbruch.

Falls Sie während der Benutzung des FileMonitor einmal

ein CLI-Fenster benötigen, können Sie vom Projektmenü aus eines öffnen. Der RUN- und der NEWCLI-Befehl müssen sich dazu jedoch in Ihrem CLI-Directory befinden. Vom Projektmenü aus kann man den FileMonitor auch beenden. Dies ist auch durch Betätigung des Close-Gadgets machbar.

## Beispiele

Um die Funktionen des FileMonitor zu überprüfen, sehen Sie in Abbildung 1 einige Beispiele anhand von Systemprogrammen. Aber wie schon gesagt, bitte nur an Kopien experimentieren, damit im Falle eines Falles die Originale noch vorhanden sind.

Bei allen Patches, bei denen Texte ersetzt werden, ist darauf zu achten, daß der Ersatztext die Länge des alten Textes nicht überschreiten darf. Ist der Ersatztext kürzer, so empfiehlt es sich, die restlichen Zeichen durch Leerzeichen zu überschreiben. Eventuell hilft es auch, eine '0' an das Ende des Ersatztextes zu hängen. Wer dies nicht beachtet, wird beim Starten des gepatchten Programms wahrscheinlich sehr bald indischen Besuch bekommen. (Viele Grüße vom Guru!) Ansonsten viel Spaß mit Ihrem neuen FileMonitor!

(jb)

```

Listings
1 ****
2 *** Filemonitor V2.0
3 *** written in 1989 by MochiSoft
4 *** (c) 1990 by DMV-Verlag
5 *** Sprache : Assembler (DEVPAC)
6 ****
7
8   opt o+.ow-
9   opt d+
10  ARP equ 1
11
12 *** Include-Dateien laden
13 incdir "RAD:include/", "SYS:include/" ;anpassen !!!
14 include exec/exec.lib
15 include exec/memory.i
16 include libraries/dos.lib
17 include libraries/dosextens.i
18 include libraries/dos.i
19 include intuition/intuition.i
20 include intuition/intuition.lib
21 include graphics/graphics.lib

```

Listing: FileMonitor.asm

```

22 include devices/console.lib
23
24 ** Konstanten aus arpbase.i
25 ArpVersion equ 34 ;Mindestversion
26 _LVOfFileRequest equ -$126
27 _LVOfBaseName equ -$276
28 _LVOfTackOn equ -$270
29 fr_Hail equ 0
30 fr_File equ 4
31 fr_Dir equ 1
32 fr_SIZEOF equ 30
33 **
34 FILENAMESIZE equ 50
35 MWidth equ 30*8+CHECKWIDTH
36 MHeight equ 1
37 MLast equ 1
38 MLeft equ 1
39 MXY equ 1
40 MItemNr equ 19
41 Leiste1Count equ 7
42 Leiste2Count equ 2
43 LAST equ -1
44
45

```

Listing: FileMonitor.asm

# Listing

```

46
47 ** WindowKonstanten **
48 WFlags equ WINDOWDEPTH!ACTIVATE!WBENCHWINDOW!WI
NDOWCLOSE!WINDOWDRAG
49 IDCMP equ REQSET!GADGETUP!MOUSEBUTTONS!CLOSEWI
NDOW!RAWKEY!MENUPICK
50
51 *** RawKeyCodes ***
52 F1_Taste equ $50
53 F2_Taste equ $51
54 F3_Taste equ $52
55 F4_Taste equ $53
56 F5_Taste equ $54
57 F6_Taste equ $55
58 F9_Taste equ $58
59 F10_Taste equ $59
60 O_Taste equ $18
61 RETURN_Taste equ $44
62 TAB_Taste equ $42
63 EESC_Taste equ $45
64 L_Taste equ $28
65 S_Taste equ $21
66 Pfeil_Hoch equ $4c
67 Pfeil_Runter equ $4d
68 Pfeil_Links equ $4f
69 Pfeil_Rechts equ $4e
70
71 *** Variablen-Struktur ***
72 rreset
73 PageStart rs.b 1
74 OldPageStart rs.b 1
75 MyIntBase rs.b 1
76 MyGfxBase rs.b 1
77 MyDosBase rs.b 1
78 MyArpBase rs.b 1
79 ConsoleBase rs.b 1
80 ReadFileHandle rs.b 1
81 WriteFileHandle rs.b 1
82 FileBuffer rs.b 1
83 MyWindow rs.b 1
84 MyFont rs.b 1
85 FileLoaded rs.w 1
86 MyUserPort rs.b 1
87 MyRastPort rs.b 1
88 AsciiOn rs.b 1
89 EditOn rs.b 1
90 Changed rs.b 1
91 HexByte rs.b 1
92 HexOffs rs.w 1
93 CrX rs.w 1
94 CrY rs.w 1
95 AltCx rs.w 1
96 AltCy rs.w 1
97 AltSecs rs.b 1
98 AltMicros rs.b 1
99 VarSize rs.b 0
100
101 *** Makros ***
102
103 Lines equ 20
104 TEXT macro
105 \1Text
106 dc.b \2
107 dc.b 0
108 \1SIZE equ *- \1Text
109 endm
110
111 # COMMAND F1_Taste,Laden
112 COMMAND macro
113 IFND COMMOUNT
114 COMMOUNT set #
115 ENDC
116 COMMOUNT set COMMOUNT+1
117 dc.w \1
118 dc.w \2Routine-Routines
119 endm
120
121 MENULEISTE macro
122 IFND MENULEISTECOUNT
123 MENULEISTECOUNT set 0
124 ENDC
125 MENULEISTECOUNT set MENULEISTECOUNT+1
126 dc.b \1SIZE*8
127 endm
128
129
130 MENU macro
131 IFND MENUCOUNT
132 MENUCOUNT set 0
133 ENDC
134 MENUCOUNT set MENUCOUNT+1
135 dc.w %11111<<11!\1<<5!ActualMenu
136 dc.w \1Routine-Routines
137 endm
138
139 LEISTE macro
140 dc.b 160,\1
141 endm
142
143 LEISTCOM macro
144 dc.w \1Routine-Routines
145 endm
146
147 * MItem Flags,ComKey,Last *,Text
148 MItem macro
149 dc.w \1!ITEMTEXT!HIGHCOMP!ITEMENABLED
150 dc.b \2

```

Listing: FileMonitor.asm

```

151 dc.b \3
152 *dc.w \4Text-Datas
153 \4 equ ItemCount
154 ItemCount set ItemCount+1
155 \4Menu equ MNR
156 MNR set MNR+1
157 endm
158
159 CALLVEC macro
160 jsr _LV0\1(a6)
161 endm
162
163 BytesPerLine equ 6*4
164 BytesPerPage equ BytesPerLine*Lines
165 TOP equ 30
166 LEFT equ 6
167 Status2Y equ TOP+Lines*8+5
168 ReqHeight equ 80
169 ReqWidth equ 300
170 ReqY equ 14
171 ReqLinks equ LEFT
172 ReqX equ (640-ReqWidth)/2
173 ReqRechts equ 640-LEFT-ReqWidth
174 *** Memory-Struktur
175 rreset
176 Vars rs.b VarSize
177 FileName rs.b FILENAMESIZE
178 AsciiSearchString rs.b FILENAMESIZE
179 HexSearchString rs.b FILENAMESIZE
180 SearchBuffer rs.b FILENAMESIZE
181 StelleAdresse rs.b 10
182 RealFileName rs.b FILENAMESIZE*2
183 UndoBuffer rs.b FILENAMESIZE*2
184 WindowTitle rs.b #
185 ReqWindowTitle rs.b #
186 PagePuffer rs.b BytesPerPage
187 Line rs.b 80
188 MyInputEvent rs.b 24
189 ToRequest rs.b 48
190 Puffer rs.b 2
191 MenuLeistenStructs rs.b mu_SIZEOF*4
192 MenuStructs rs.b (it_SIZEOF+mi_SIZEOF)*MitemNr
193 WindowInfo rs.b nw_SIZE
194 MyStrGadget rs.b si_SIZEOF
195 MyBorder rs.b bd_SIZEOF
196 MyCursor rs.b it_SIZEOF
197 Dateizeigertext rs.b 44
198 DosErrorText rs.b 22
199 LInfoCoords1 rs.w Leiste1Count+1
200 LInfoCoords2 rs.w Leiste2Count+1
201 MyRequester rs.b rq_SIZEOF
202 ReqBorder rs.b bd_SIZEOF
203 MyGadget1 rs.b gg_SIZEOF
204 MyGadget2 rs.b gg_SIZEOF
205 MyGadget3 rs.b gg_SIZEOF
206 MyGadget4 rs.b gg_SIZEOF
207 MyBorder2 rs.b bd_SIZEOF
208 MyBorder3 rs.b bd_SIZEOF
209 MyBorder4 rs.b bd_SIZEOF
210 MyText1 rs.b it_SIZEOF
211 MyText2 rs.b it_SIZEOF
212 MyText3 rs.b it_SIZEOF
213 MyText4 rs.b it_SIZEOF
214 ReqText1 rs.b it_SIZEOF
215 ReqText2 rs.b it_SIZEOF
216 ReqText3 rs.b it_SIZEOF
217 CancelText rs.b it_SIZEOF
218 MemXY1 rs.w 10
219 MemXY2 rs.w 10
220 MemXY3 rs.w 10
221 MemXY4 rs.w 10
222
223 IFD ARP
224 FileReq rs.b fr_SIZEOF
225 DirName rs.b FILENAMESIZE*2
226 ENDC
227 GetMem rs.b #
228
229 Mem equur a4
230 Var equur Mem
231 FileSize equur a5
232
233 ***** Programm *****
234
235 *** Workbench od. CLI ?
236 move.l d0,d7 ;Parameter
237 move.l a0,a5 ;retten
238 and.w #31,d7 ;Parameterlaenge
239 begrenzen
240 move.l (_SysBase).w,a6
241 suba.l a1,a1
242 CALLVEC FindTask
243 move.l d0,a2
244 tst.l pr_CLI(a2)
245 bne.s Program
246 lea pr_MsgPort(a2),a0
247 CALLVEC WaitPort
248 lea pr_MsgPort(a2),a0
249 CALLVEC GetMsg
250 move.l d0,-(sp)
251 moveq #1,d7 ;Parameterlaenge
252 bsr.s Program
253 CALLVEC Forbid
254 move.l (sp)+,a1
255 jmp _LV0ReplyMsg(a6)

```

Listing: FileMonitor.asm



```

256 Program
257 * Speicher holen
258 move.l #GetMem,d0
259 move.l #MEMF_PUBLIC!MEMF_CLEAR,d1
260 CALLVEC AllocMem
261 tst.l d0
262 beq MemErr
263 move.l d0,Mem
264 * Strukturen initialisieren
265 lea MenuTexts(pc),a3
266 lea MenuLeisten(pc),a0
267 lea MenuLeistenStructs(Mem),a1
268 bsr GetMenuLeisten ;Menus
269 lea MenuLeistenStructs(Mem),a0
270 lea MenuItem(pc),a1
271 lea MenuItemStructs(Mem),a2
272 bsr GetMenu ;Menuitems
273 lea MyGadget1+4(Mem),a0 Gadget
274 move.l #14<<16+30,(a0)+
275 move.l #14<<16+30,(a0)+
276 addq.l #2,a0
277 move.l #14<<16+30,(a0)+
278 ET,(a0)+
279 move.l #49<<16+49,si_BufferPos+MyStrGadget(Mem)
280 lea UndoBuffer(Mem),a0
281 move.l a0,MyStrGadget+si_UndoBuffer(Mem)
282 moveq #1,d0
283 lea WindowInfo+4(Mem),a0 ;Window
284 move.l #640<<16+200,(a0)+
285 move.w d0,(a0)+
286 move.l #IDCMP,(a0)+
287 move.l #WFLags,(a0)+
288 lea WindowName(pc),a1
289 move.l a1,nw_Title+WindowInfo(Mem)
290 move.w d0,nw_Type+WindowInfo(Mem)
291 lea MyBorder(Mem),a0 ;Border
292 move.l #LEFT<<16+TOP,(a0)+
293 move.l #3<<24+$0205,(a0)+
294 lea XY(pc),a0 ;XY
295 lea MemXY1(Mem),a1
296 moveq #4*10-1,d0
297 InitXYLoop
298 move.b (a0)+,d1
299 ext.w d1
300 move.w d1,(a1)+
301 dbf d0,InitXYLoop
302 lea MyCursor(Mem),a0 ;InituiText
303 move.l #03000800,(a0)+
304 move.l #LEFT<<16+TOP-6,(a0)+
305 lea DateZeigerText(Mem),a1
306 lea ZeigerText(pc),a0
307 CopyZeigerText
308 move.b (a0)+,d0
309 move.b d0,(a1)+
310 addq.b #2,d0 ;+2?
311 bne.s CopyZeigerText
312 lea MemXY4(Mem),a0
313 lea MyBorder2(Mem),a1
314 moveq #2,d1
315 InitBorder
316 move.l Bba0,bd_XY(a1)
317 move.l #03000005,bd_FrontPen(a1)
318 lea bd_SIZEOF(a1),a1
319 dbf d1,InitBorder
320 lea MyBorder4(Mem),a1
321 move.l a1,MyBorder3+bd_NextBorder(Mem)
322 lea MemXY3(Mem),a0
323 move.l a0,bd_XY(a1)
324 moveq #2,d0
325 lea ReqText1(Mem),a0
326 move.l #14<<16+20,d1
327 InitCoord
328 move.b d0,(a0)
329 move.l d1,lt_LeftEdge(a0)
330 lea lt_SIZEOF(a0),a0
331 subq.w #8,d1
332 bpl.s InitCoord
333 * dos.library oeffnen
334 IFD ARP
335 lea ArpNm(pc),a1
336 moveq #ArpVersion,d0
337 CALLVEC OpenLibrary
338 move.l d0,MyArpBase(Mem)
339 ENDC
340 lea DosName(pc),a1
341 CALLVEC OldOpenLibrary
342 move.l d0,MyDosBase(Var)
343 * intuition.library oeffnen
344 lea IntName(pc),a1
345 moveq #33,d0 ;Kick 1.2
346 CALLVEC OpenLibrary
347 move.l d0,MyIntBase(Var)
348 beq IntErr
349 * graphics.library oeffnen
350 lea GfxName(pc),a1
351 CALLVEC OldOpenLibrary
352 move.l d0,MyGfxBase(Var)
353 * Window oeffnen
354 move.l MyIntBase(Var),a6
355 lea WindowInfo(Mem),a0
356 CALLVEC OpenWindow
357 move.l d0,MyWindow(Var)
358 beq WindowErr
359 move.l d0,a2
360 lea MenuLeistenStructs(Mem),a1
361 move.l a2,a0

```

Listing: FileMonitor.asm

```

361 CALLVEC SetMenuStrip
362 move.l wd_UserPort(a2),MyUserPort(Var)
363 move.l wd_RPort(a2),MyRastPort(Var)
364 move.l wd_WScreen(a2),a0
365 cmp.w #212,sc_Height(a0)
366 blo.s NSize
367 move.l a2,a0
368 moveq #0,d0
369 moveq #12,d1
370 CALLVEC MoveWindow
371 NSize
372 * Topaz.Font (8) holen
373 move.l MyGfxBase(Var),a6
374 lea MyTextAttr(pc),a0
375 CALLVEC OpenFont
376 move.l d0,MyFont(Var)
377 move.l MyRastPort(Var),a1
378 move.l d0,a0
379 * Und setzen
380 CALLVEC SetFont
381 moveq #3,d0
382 bsr Pen
383 moveq #TOP-10,d1
384 bsr DrawHori
385 lea GadgetLeiste1+1(pc),a3
386 moveq #TOP-12,d3
387 move.w #638-Leiste1Size*8+8,d0
388 moveq #Leiste1Size-1,d1
389 bsr Textout3
390 lea GadgetLeiste2(pc),a3
391 move.w #Status2Y,d3
392 move.w #638-Leiste2Size*8,d0
393 moveq #Leiste2Size,d1
394 bsr Textout3
395 moveq #10,d1
396 lea GadgetLeiste1+Leiste1Size(pc),a0
397 lea LInfoCoords1+Leiste1Count*2(Mem),a1
398 moveq #Leiste1Count-1,d0
399 bsr GetLeiste
400 bsr DrawHori
401 move.w #TOP+Lines*8-3,d1
402 lea GadgetLeiste2+Leiste2Size(pc),a0
403 lea LInfoCoords2+Leiste2Count*2(Mem),a1
404 moveq #Leiste2Count,d0
405 bsr GetLeiste
406 bsr DrawHori
407 move.w #198,d1
408 bsr DrawHori
409 * console - library oeffnen
410 lea ConsoleName(pc),a0
411 lea IoRequest(Mem),a1
412 moveq #-1,d0
413 moveq #0,d1
414 move.l (_SysBase).w,a6
415 CALLVEC OpenDevice
416 tst.l d0
417 bne DeviceError
418
419 move.l 20+IoRequest(Mem),ConsoleBase(Var)
420
421 subq.w #2,d7
422 bml.s DoLoad
423 cmp.b #10,(a5)
424 beq.s DoLoad
425 lea RealFileName(Mem),a0
426 CopyParam
427 move.b (a5)+,(a0)+
428 dbf d7,CopyParam
429 clr.b (a0)
430 bsr LOADRoutine2
431 bra.s HoleMsg
432 DoLoad bsr DoBorder
433 moveq #0,d6
434 bsr LOADRoutine
435 *** Hauptschleife ***
436 HoleMsg
437 * Auf Message warten
438
439 move.l (_SysBase).w,a6
440 move.l MyUserPort(Var),a0
441 CALLVEC WaitPort
442 move.l MyUserPort(Var),a0
443 CALLVEC GetMsg
444 move.l d0,a0
445 move.l a0,a1
446 lea 20(a0),a0
447 * Wichtige Werte retten
448 move.l (a0)+,d4 class
449 move.w (a0)+,d5 code
450 move.w (a0)+,d6 qual
451 addq.l #5,a0
452 move.w (a0)+,d7 mousex
453 swap d7
454 move.w (a0)+,d7 mousey
455 move.l (a0)+,d2 secs
456 move.l (a0)+,d3 micros
457 * InputEvent rekonstruieren
458 cmp.w #RAWKEY,d4
459 bne.s NCon21e
460 move.b #1,4+MyInputEvent(Mem) RAWKEY
461 move.w d5,MyInputEvent+6(Mem) CODE
462 move.w d6,MyInputEvent+8(Mem) Qual
463 NCon21e
464 * Message beantworten
465 CALLVEC ReplyMsg
466 cmp.w #CLOSEWINDOW,d4

```

Listing: FileMonitor.asm

```

467 beq Quit
468 cmp.w #MOUSEBUTTONS,d4
469 bne NKlick
470 cmp.w #SELECTDOWN,d5
471 GoNE_NextMsg
472 bne.s HoleMsg
473 tst.b FileLoaded(Var)
474 beq.s NL2
475 cmp.w #TOP-10,d7
476 bhi.s NL1
477 lea L1Info(pc),a1
478 lea L1InfoCoords1(Mem),a0
479 swap d7
480 bsr TestLeiste
481 bra ClearHexBuf
482 NL1 cmp.w #TOP+Lines*8-3,d7
483 blo.s NL2
484 swap d7
485 lea L1InfoCoords2(Mem),a0
486 lea L2Info(pc),a1
487 bsr TestLeiste
488 bra ClearHexBuf
489 NL2
490 move.w d7,d1
491 swap d7
492 sub.w #TOP-6,d1
493 blo NLocate
494 cmp.w #Lines*8-1,d1
495 bhi.s NLocate
496 move.w d7,d0
497 sub.w #LEFT+8*(BytesPerLine*2+BytesPerLine/4),d0
498 blo.s NAscii
499 lsr.w #3,d0
500 st.b d7
501 DoLocate
502
503 lsr.w #3,d1
504 cmp.w #BytesPerLine-1,d0
505 bhi.s GoNE_NextMsg
506 move.l AltSecs(Var),d4
507 move.l AltMicros(Var),d5
508 move.l d2,AltSecs(Var)
509 move.l d3,AltMicros(Var)
510 cmp.w AltCx(Var),d0
511 bne.s DoLocate2
512 cmp.w AltCy(Var),d1
513 bne.s DoLocate2
514 move.l d4,d0
515 move.l d5,d1
516 move.l MyIntBase(Var),a6
517 CALLVEC DoubleClick
518 tst.b d0
519 beq.s Go_NextMsg
520 bsr SetEditMode
521 Go_NextMsg
522 bra NextMsg
523 DoLocate2
524
525 bsr CursorOff
526 move.w d0,CrX(Var)
527 move.w d1,CrY(Var)
528 bsr CursorOn
529 bra.s ClearHexBuf
530 NAscii move.w d7,d0
531 sub.w #LEFT-4,d0
532 blo.s NLocate
533 lsr.w #3,d0
534 move.w d0,d6
535 ext.l d6
536 divu.w #9,d6
537 sub.w d6,d0
538 swap d6
539 tst.w d6
540 beq.s NLocate
541 subq.w #1,d0
542 lsr.w #1,d0
543 sf.b d7
544 bra.s DoLocate
545
546 NAtHex
547 NLocate
548 NKlick
549 cmp.w #MENU PICK,d4
550 bne.s NoMenu
551 lea Menu(pc),a0
552 moveq #MENU COUNT-1,d0
553 SearchMenu
554 cmp.w (a0),d5
555 addq.l #4,a0
556 dbeq d0,SearchMenu
557 bne.s Go_NextMsg
558 bra.s StartComm
559 NoMenu
560 * Auf Funktion der Taste ueberpruefen
561 cmp.w #RAWKEY,d4
562 bne NoRawKey
563 tst.b FileLoaded(Var)
564 beq NoRawKey
565 lea Commands(pc),a0
566 moveq #COMM COUNT-1,d0
567 SearchComm
568 cmp.w (a0),d5
569 addq.l #4,a0
570 dbeq d0,SearchComm
571 bne.s NoSub
572 StartComm

```

Listing: FileMonitor.asm

```

573 lea Routines(pc),a1
574 add.w -(a0),a1
575 jsr (a1) ;Funktion anspr
ngen
576 ClearHexBuf
577 pea NextMsg(pc)
578 bra ClearHexBufSub
579 NoSub
580 move.w CrX(Var),d2
581 bsr CheckLimits
582 beq NoRawKey
583 * Editieren
584 tst.b EditOn(Var)
585 beq NoRawKey
586 move.l ConsoleBase(Var),a6
587 lea MyInputEvent(Mem),a0
588 lea Puffer(Mem),a1
589 suba.l a2,a2
590 moveq #1,d1
591 CALLVEC RawKeyConvert ;Asciwert holen
592 tst.w d0
593 ble NoRawKey
594 move.b Puffer(Mem),d5
595 cmp.b #32,d5
596 blo NoRawKey
597 move.w CrY(Var),-(sp)
598 move.w CrX(Var),-(sp)
599 move.l PageStart(Var),-(sp)
600 tst.b AsciiOn(Var)
601 bne.s NHexEd
602 * Hex-Seite editieren
603 bsr UpperD5
604 sub.b #'0',d5
605 bml.s CharOverflow
606 cmp.b #9,d5
607 bls.s NGreaterDez
608 subq.b #'A'-'9'-1,d5
609 NGreaterDez
610 cmp.b #F,d5
611 bhi.s CharOverflow
612 move.w HexOffs(Var),d3
613 subq.w #1,HexOffs(Var)
614 lsl.w #2,d3
615 and.b #F,d5
616 lsl.w d3,d5
617 or.b d5,HexByte(Var)
618 move.b HexByte(Var),d5
619 tst.w d3
620 bne.s CharOverflow
621 pea NHexEd(pc)
622 ClearHexBufSub
623 move.w #1,HexOffs(Var)
624 clr.b HexByte(Var)
625 rts
626 CharOverflow
627 move.l (sp)+,a0
628 movem.w (sp)+,d0-d1
629 bra.s CharOverflow2
630 NHexEd
631 bsr CRghtRoutine
632 bsr CursorOn
633 move.l (sp)+,a0
634 cmpa.l PageStart(Var),a0 gescrollt?
635 beq.s NAdaptY
636 subq.w #1,2(sp)
637 NAdaptY
638 movem.w (sp),d0-d1
639 bsr PageOffset2
640 move.l PageStart(Var),a2
641 move.b d5,(a2,d0)
642 st.b Changed(Mem)
643 * aktuelle Zeile neu anzeigen
644 ShowLine
645 adda.w d1,a2
646 movem.w (sp)+,d0-d1
647 move.l MyGfxBase(Var),a6
648 move.w d1,d0
649 lsl.w #3,d0
650 add.w #TOP,d0
651 bsr LineLocate
652 bsr ConvertLine
653 bsr DoBorder
654 CharOverflow2
655 NoRawKey
656 NextMsg
657 bra HoleMsg
658 Quit
659 * Datei schliessen, Dateispeicher freigeben
660 bsr LooseChangesTest
661 beq.s NextMsg
662 bsr CloseDatei
663 move.l (_SysBase).w,a6
664 lea IORequest(Mem),a1
665 CALLVEC CloseDevice
666 DeviceError
667 * Window weg!
668 move.l MyFont(Var),a1
669 move.l MyGfxBase(Var),a6
670 CALLVEC CloseFont
671 move.l MyIntBase(Var),a6
672 move.l MyWindow(Var),a0
673 CALLVEC ClearMenuStrip
674 move.l MyWindow(Var),a0
675 CALLVEC CloseWindow
676 WindowErr
677 * GFX weg

```

Listing: FileMonitor.asm



```

678 move.l (_SysBase).w,a6
679 move.l MyGfxBase(Var),a1
680 CALLVEC CloseLibrary
681 * Intuition schliessen
682 move.l MyIntBase(Var),a1
683 CALLVEC CloseLibrary
684 IntErr
685 * No more DOS!
686 move.l MyDosBase(Var),a1
687 CALLVEC CloseLibrary
688 IFD ARP
689 move.l MyArpBase(Mem),d0
690 beq.s NArpClose
691 move.l d0,a1
692 CALLVEC CloseLibrary
693 NArpClose
694 ENOC
695 * Free the Mem!
696 move.l Mem,a1
697 move.l @GetMem,d0
698 CALLVEC FreeMem
699 moveq #0,d0
700 MemErr rts ;Hello EXEC, I'm
    ready!

701
702
703 *** Tasten-Routinen ***
704 Routines
705 *** Cursor Hoch ***
706 CupRoutine
707 bsr.s AltTest
708 bne FILESTARTRoutine
709 bsr.s ShiftTest
710 beq.s NShift
711 PAGEUPRoutine
712 move.l PageStart(Var),a2
713 suba.w #BytesPerPage,a2
714 move.w CrX(Var),d2
715 ext.l
716 cmpa.l FileBuffer(Var),a2
717 blo FileLocate
718 bsr CursorOff
719 move.l a2,PageStart(Var)
720 bsr ZeigeSeite
721 bra.s ReturnSub7

```

Listing: FileMonitor.asm

```

722 NShift
723 Cup2Routine
724 tst.w CrY(Var)
725 bne.s NScrUp
726 bsr CursorOff
727 moveq #-BytesPerLine,d2
728 bsr CheckLimits
729 beq.s NUP
730 moveq #-8,d1
731 moveq #TOP,d0
732 bsr Scroll
733
734 move.l PageStart(Var),a2
735 adda d2,a2
736 move.l a2,PageStart(Var)
737 bsr ConvertLine
738 bsr PufDown
739 NUP
740 bsr CursorOn
741 bra.s ReturnSub7
742 NScrUp
743 subq.w #1,CrY(Var)
744 bra Cursor
745
746 * In: Qualifier in d6, Out: Z=1 wenn Alt *
747 AltTest move.b d6,d0
748 and.b #X110000,d0
749 ReturnSub7
750 rts
751
752 * In: Qualifier in d6, Out: Z=1 wenn Shift *
753 ShiftTest
754 move.b d6,d0
755 and.b #X11,d0
756 rts
757
758 *** Cursor unten ***
759 CDownRoutine
760 bsr.s AltTest
761 bne FILEENDRoutine
762 bsr.s ShiftTest
763 beq.s NShiftDown
764 PAGEDOWNRoutine
765 move.l PageStart(Var),a2
766 adda.w #BytesPerPage,a2

```

Listing: FileMonitor.asm

# Bilder wie im Original!

In Sekundenbruchteilen von Video und Kamera eingelesen.

## SNAPSHOT!

ECHTZEIT-  
VIDEO-DIGITIZER

unentbehrlich  
bei:

- Grafik
- Animation
- Desktop Video
- Präsentation

Komplett-Set



Black-White-Digitizer



Farb-Adapter



Herstellung und Vertrieb  
sowie kostenlose Informationen bei:

# VIDEOTECHNIK DIEZEMANN

Postfach 4365 · Dammstraße 42 · 2300 Kiel 1 · Telefon (0431) 9 44 24 · Telefax (0431) 9 24 32

```

767     move.l    a2,a1
768     *add.w    #BytesPerPage,a1
769     suba.l    FileBuffer(Var),a1
770     cmpa.l    FileSize,a1
771     bhi       FILEENDRoutine
772     bsr       CursorOff
773     move.l    a2,PageStart(Var)
774     bsr       ZeigeSeite
775     bra.s     NDown2
776 NShiftDown
777 CDown2Routine
778     cmp.w     #Lines-1,CrY(Var)
779     bne.s     NScrDown
780     bsr       CursorOff
781     move.w     #BytesPerLine,d2
782     bsr.s     CheckLimits
783     beq.s     NDown
784     move.w     #-BytesPerPage,d2
785     bsr.s     CheckLimits
786     moveq     #8,d1
787     move.w     #TOP+(Lines-1)*8,d0
788     bsr       Scroll
789
790     move.l     PageStart(Var),a2
791     lea        BytesPerPage(a2),a2
792     bsr       ConvertLine
793
794     lea        -BytesPerPage(a2),a2
795     move.l     a2,PageStart(Var)
796     bsr       PufUp
797 NDown
798     bsr.s     CursorOn
799     bra.s     NDown2
800 NScrDown
801     addq.w     #1,CrY(Var)
802     bsr.s     Cursor
803 NDown2
804     rts
805 *** Cursor Links ***
806 CLftRoutine
807     subq.w     #1,CrX(Var)
808     *st.w     CrX(Var)
809     bpl.s     Cursor
810     move.w     #BytesPerLine-1,CrX(Var)
811     bra       CUp2Routine
812
813 *** Cursor Rechts ***
814 CRghtRoutine
815     addq.w     #1,CrX(Var)
816     cmp.w     #BytesPerLine,CrX(Var)
817     bne.s     Cursor
818     clr.w     CrX(Var)
819     bra       CDown2Routine
820
821 QUITRoutine
822     addq.l     #4,sp
823     bra       Quit
824
825 ** d2=Offset zu altem Wert * Z=1 wenn Error **
826 CheckLimits
827     move.l     PageStart(Var),a2
828     bsr.s     PageOffset
829     add.w     d2,d0
830     adda.w     d0,a2
831     move.l     FileBuffer(Var),a0
832     cmpa.l     a0,a2
833     bml.s     Overflow
834     move.l     FileSize,a1
835     adda.w     CrX(Var),a1
836     adda.l     a1,a0
837     cmp.l     a0,a2
838     blo.s     ExitCheck
839 Overflow
840     moveq     #0,d0
841 ExitCheck
842     rts
843
844 * aus CrX u. CrY Offset zum ersten Byte der Seite nach 0
845
846 PageOffset
847     move.w     CrX(Var),d0
848     move.w     CrY(Var),d1
849 PageOffset2
850     mulu       #BytesPerLine,d1
851     add.w     d1,d0
852     ext.l     d0
853     rts
854
855 *** Cursor an/aus Routine
856 Cursor
857     bsr.s     DoBorder
858 CursorOn
859     move.l     PageStart(Var),OldPageStart(Var)
860     move.w     CrX(Var),AltCx(Var)
861     move.w     CrY(Var),AltCy(Var)
862     movem.l    d0-d1/a1/a0/a6,-(sp)
863     lea        FilePosText-ZeigerText+Dateizeigertext(Mem),
a0
864     bsr.s     PageOffset
865     move.l     PageStart(Var),d1
866     add.l     d1,d0
867     sub.l     FileBuffer(Var),d0
868     bsr       LONGTOHEX
869     move.l     MyGfxBase(Var),a6
870     move.l     MyRastPort(Var),a1

```

Listing: FileMonitor.asm

```

871     moveq     #5+(dateizeigende-dateizeig)*8,d0
872     move.w     #Status2Y,d1
873     CALLVEC    Move
874     bsr       Pen1
875     move.l     MyRastPort(Var),a1
876     lea        FilePosText-ZeigerText+Dateizeigertext(Mem),
a0
877     moveq     #8,d0
878     CALLVEC    Text
879     movem.l    (sp)+,d0-d1/a1/a0/a6
880 CursorOff
881 DoBorder
882     movem.l    d0-d2/a0-a2/a6,-(sp)
883     move.w     AltCx(Var),d0
884     bsr.s     GetHexx
885     lea        MemXY1(Mem),a2
886     move.l     MyIntBase(Var),a6
887     bsr.s     DoDrawBorder
888     lea        MemXY2(Mem),a2
889     move.w     AltCx(Var),d0
890     lsl.w     #3,d0
891     add.w     #6*8,d0
892     bsr.s     DoDrawBorder
893     bsr.s     EditCursor
894     movem.l    (sp)+,d0-d2/a0-a2/a6
895     rts
896
897 DoDrawBorder
898     move.w     AltCy(Var),d1
899     lsl.w     #3,d1
900     lea        MyBorder(Mem),a1
901     move.l     a2,bd_XY(a1)
902     move.l     MyRastPort(Var),a0
903     jmp        _LV0DrawBorder(a6)
904
905 GetHexx     move.w     d0,d2
906     lsl.w     #1,d0
907     *ext.l    d2
908     *divu.w   #4,d2
909     lsr.w     #2,d2
910     add.w     d2,d0
911     lsl.w     #3,d0
912     rts
913
914
915 * Scrolle Ausschnitt vertikal um D1
916 Scroll
917     move.l     MyGfxBase(Var),a6
918     movem.l    d0-d5/a1,-(sp)
919     moveq     #0,d0
920     move.l     MyRastPort(Var),a1
921     moveq     #LEFT,d2
922     moveq     #TOP-6,d3
923     move.w     #LEFT+78*8,d4
924     move.w     #TOP+(Lines-1)*8+1,d5
925     CALLVEC    ScrollRaster
926     movem.l    (sp)+,d0-d5/a1
927
928 * GFX-Cursor == LEFT,Y=d0 setzen
929 LineLocate
930     move.w     d0,d1
931     move.l     MyRastPort(Var),a1
932     moveq     #LEFT,d0
933     jmp        _LV0Move(a6)
934
935 *** EditCursor an/aus Routinen
936 EditCursor2
937     move.w     CrX(Var),d0
938     move.w     CrY(Var),d1
939     bra.s     EditCursor3
940 EditCursor
941     tst.b     EditOn(Var)
942     beq.s     NoEditCur
943     move.w     AltCy(Var),d1
944     move.w     AltCx(Var),d0
945 EditCursor3
946     lsl.w     #3,d1
947     tst.b     AsciiOn(Var)
948     beq.s     HexOn
949     lea        Space(pc),a0
950     lsl.w     #3,d0
951     add.w     #BytesPerLine*2+BytesPerLine/4*8,d0
952     bra.s     EditCurOut
953 HexOn
954     lea        DoubleSpace(pc),a0
955     bsr.s     GetHexx
956
957 EditCurOut
958     lea        MyCursor(Mem),a1
959     move.l     a0,it_IText(a1)
960     move.l     MyRastPort(Var),a0
961     move.l     MyIntBase(Var),a6
962     CALLVEC    PrintIText
963 NoEditCur
964 ReturnSub10
965     rts
966
967 *** Neue Datei laden
968 LOADRoutine
969     bsr       LooseChangesTest
970     beq.s     ReturnSub10
971     bsr       CursorOff
972     lea        LadenFrage(pc),a0
973     clr.b     IsFileReq(Mem)
974     bsr       GetFileName
975     beq       NichtLaden2

```

Listing: FileMonitor.asm



```

976 LOADRoutine2
977     bsr     CloseDatei
978     lea     RealFileName(Mem),a0
979     move.l  a0,d1
980     move.l  MyDosBase(Var),a6
981     move.l  #MODE_OLDFILE,d2
982     CALLVEC Open
983     move.l  d0,ReadFileHandle(Var)
984     bhl.s   Load
985     bsr     IO_Error_Out
986     bra.s   NichtLaden
987 Load
988     bsr     GetSize
989     move.l  (_SysBase).w,a6
990     move.l  FileSize,d0
991     add.l   #BytesPerPage,d0
992     move.l  #$10000,d1
993     CALLVEC AllocMem
994     move.l  d0,FileBuffer(Var)
995     bne.s   MemOk
996     lea     LowMemText(pc),a3
997     moveq   #1,d0
998     bsr     MessageWindow
999     bra.s   NichtLaden
1000 MemOk
1001     move.l  FileBuffer(Var),a0
1002     adda.l  FileSize,a0
1003     move.w  #BytesPerPage-1,d1
1004 EndCharLoop
1005     move.b  #"",(a0)+
1006     dbf     d1,EndCharLoop
1007     move.l  FileBuffer(Var),a2
1008     move.l  a2,PageStart(Var)
1009     bsr     LoadBuffer
1010     move.l  d0,-(sp)
1011     move.l  ReadFileHandle(Var),d1
1012     CALLVEC Close
1013     cmp.l   (sp)+,FileSize
1014     beq.s   NLoadError
1015     bsr     CloseDatei
1016 NLoadError
1017 NichtLaden
1018     bsr.s   ShowTitle
1019     lea     FileLenText-ZeigerText+Dateizeigertext(Mem),
a0
1020     move.l  FileSize,d0
1021     bsr     LONGToHEX
1022 NichtLaden2
1023     bsr     Pen1
1024     move.l  MyWindow(Var),a2
1025     lea     Dateizeigertext(Mem),a3
1026     move.w  #Status2Y,d3
1027     moveq   #ZeigerTextEnde-ZeigerText,d1
1028     move.l  MyGfxBase(Var),a6
1029     bsr     Textout2
1030     bra     ZeigeSeiteImmer
1031
1032 ***** WindowTitel zeigen! *****
1033 ShowTitle
1034     lea     WindowTitle(Mem),a0
1035     move.l  a0,a1
1036     lea     RealFileName(Mem),a2
1037     moveq   #78/2-1,d0
1038 CopyTitle
1039     move.w  (a2)+,(a0)+
1040     dbf     d0,CopyTitle
1041     move.l  MyWindow(Var),a0
1042 ShowTitle2
1043     tst.l   FileBuffer(Var)
1044     bne.s   ShowFileName
1045     lea     WindowName(pc),a1
1046 ShowFileName
1047     lea.l   ScreenTitle(pc),a2
1048     move.l  MyIntBase(Var),a6
1049     jmp     _LV0SetWindowTitles(a6)
1050
1051
1052
1053 *** Datei speichern ***
1054 SAVERoutine
1055     tst.b   FileLoaded(Var)
1056     bne.s   FileIsThere
1057     lea     NoFileText(pc),a3
1058     moveq   #0,d0
1059     bra     MessageWindow
1060 FileIsThere
1061     lea     SpeichernFrage(pc),a0
1062     bset    #5,16+FileReq(Mem)
1063     bsr     GetFileName
1064     beq.s   Nichtschreiben
1065     lea     RealFileName(Mem),a2
1066     move.l  a2,d1
1067     move.l  a2,d4
1068     move.l  #ACCESS_READ,d2
1069     move.l  MyDosBase(Var),a6
1070     CALLVEC Lock
1071     move.l  d0,d1
1072     beq.s   NoOldFile
1073     CALLVEC UnLock
1074     lea     SaveVerifyText(pc),a3
1075     bsr     YesNoRequest
1076     beq.s   ReturnSub8
1077 NoOldFile
1078     move.l  d4,d1
1079     move.l  #MODE_NEWFILE,d2
1080     CALLVEC Open

```

Listing: FileMonitor.asm

```

1081     move.l  d0,WriteFileHandle(Var)
1082     bhl.s   Save
1083     bsr     IO_Error_Out
1084     bra.s   Nichtschreiben
1085 Save
1086     move.l  FileBuffer(Var),a0
1087     move.l  a0,d2
1088     move.l  FileSize,d3
1089     move.l  WriteFileHandle(Var),d1
1090     CALLVEC Write
1091     move.l  d0,d5
1092     bpl.s   CloseIt
1093     bsr     IO_Error_Out
1094     bsr.s   CloseIt
1095     move.l  d4,d1
1096     jmp     _LV0DeleteFile(a6)
1097
1098 CloseIt
1099     move.l  WriteFileHandle(Var),d1
1100     CALLVEC Close
1101     tst.l   d5
1102     bml.s   ReturnSub8
1103     bsr     ShowTitle
1104     sf.b    Changed(Var)
1105 Nichtschreiben
1106 ReturnSub8
1107     rts
1108
1109 WriteError
1110
1111
1112 *** Seitenpuffer zurueckschreiben
1113 RESTORERoutine
1114     tst.b   FileLoaded(Var)
1115     beq.s   ReturnSub8
1116     bsr     CopyToPage
1117     bsr     CursorOff
1118     bra     ZeigeSeiteImmer
1119
1120
1121 *** AsciiString suchen
1122 ASCIIPREVRoutine
1123     moveq   #-1,d7
1124     bra.s   StartSearch
1125 ASCIIFINDRoutine
1126     lea     ASCII_suchText(pc),a0
1127     move.w  #ReqRechts,d2
1128     bsr     GetString2
1129     bhl.s   ASCIIPREVRoutine
1130     beq.s   ReturnSub8
1131 ASCIIINEXTRoutine
1132     moveq   #1,d7
1133 StartSearch
1134     tst.b   AsciiSearchString(Mem)
1135     beq.s   ReturnSub8
1136     bsr     CursorOff
1137     lea     AsciiSearchString(Var),a1
1138     moveq   #0,d4 ;laenge
1139 AsciiLenLoop
1140     tst.b   (a1,d4)
1141     beq.s   LenFound
1142     addq.l  #1,d4
1143     bra.s   AsciiLenLoop
1144 LenFound
1145     tst.b   d7
1146     bpl.s   n
1147     neg.l   d4
1148     subq.l  #1,a1
1149     sub.l   d4,a1
1150 n
1151     move.l  MenuLeistenStructs+1*mu_SIZEOF+mu_FirstItem(
Mem),a0
1152     btst    #0,mi_Flags(a0)
1153     seq.b   d3
1154 SEARCH
1155     bsr     PageOffset
1156     ext.l   d0
1157     move.l  PageStart(Var),a0
1158     add.l   a0,d0
1159     move.l  FileBuffer(Var),a0
1160     sub.l   a0,d0
1161     moveq   #0,d1
1162     tst.b   d7
1163     bpl.s   n3
1164     sub.l   d4,d0
1165     cmp.l   FileSize,d0
1166     bls.s   ngt
1167     move.l  FileSize,d0
1168 ngt
1169     bra.s   AsciiSearch
1170 n3
1171     addq.l  #1,d0 ;erstes ueberspringen
1172 AsciiSearch
1173     sub.l   d1,d0
1174     move.l  d7,d1
1175     neg.l   d1
1176 AsciiSearch2
1177     cmp.l   FileSize,d0
1178     bhl     NoLocate
1179     add.l   d7,d1
1180     cmp.w   d4,d1
1181     beq.s   ASCIIIFound
1182     add.l   d7,d0
1183     bml     NoLocate
1184     move.b  -(a0,d0.1),d2
1185     move.b  (a1,d1.1),d5

```

Listing: FileMonitor.asm

# Listing

```

1186 tst.b d3
1187 bne.s CaseIsAn
1188 cmp.b #'a',d2
1189 blo.s NUpper
1190 bclr #5,d2
1191 NUpper bsr.s UpperD5
1192 CaseIsAn
1193 cmp.b d5,d2
1194 bne.s AsciiSearch
1195 bra.s AsciiSearch2
1196 ASCIIFound
1197 sub.l d1,d0
1198 tst.b d7
1199 bpl.s n2
1200 add.l d1,d0
1201 subq.l #1,d0
1202 n2 move.l d0,d2
1203 bra FileLocate2
1204
1205 UpperD5
1206 cmp.b #'a',d5
1207 blo.s NUpper2
1208 bclr #5,d5
1209 ReturnSub9
1210 NUpper2 rts
1211
1212 * HexString suchen
1213 HEXPREVRoutine
1214 moveq #-1,d7
1215 bra.s HEXStart
1216 HEXFINDRoutine
1217 lea HEX_suchText(pc),a0
1218 moveq #ReqLinks,d2
1219 bsr GetString3
1220 beq.s ReturnSub9
1221 bhi.s HEXPREVRoutine
1222 HEXNEXTRoutine
1223 moveq #1,d7
1224 HEXStart
1225 tst.b HexSearchString(Mem)
1226 beq.s ReturnSub9
1227 bsr CursorOff
1228 lea HexSearchString(Mem),a2
1229 lea SearchBuffer(Mem),a1
1230 moveq #-1,d4 laenge
1231 ConvfromHex
1232 addq.l #1,d4
1233 moveq #0,d2 ergebnisByte
1234 moveq #4,d3 shiftwert
1235 GetHexByte
1236 tst.w d3
1237 bml.s ConvfromHex
1238 move.b (a2)+,d5
1239 beq.s Convready
1240 bsr.s UpperD5 UpperCase
1241 sub.b #'0',d5
1242 bml.s HexStringErr
1243 cmp.b #9,d5
1244 bls.s NGreaterDez2
1245 subq.b #'A'-'9'-1,d5
1246 NGreaterDez2
1247 cmp.b #$F,d5
1248 bhi.s HexStringErr
1249 lsl.w d3,d5
1250 or.b d5,d2
1251 subq.l #4,d3
1252 move.b d2,(a1,d4,1)
1253 bra.s GetHexByte
1254
1255 Convready
1256 tst.b d7
1257 bpl.s n4
1258 neg.l d4
1259 subq.l #1,a1
1260 sub.l d4,a1
1261 n4
1262 cmp.w #4,d3
1263 beq SEARCH
1264 HexStringErr
1265 lea.l HEX_ErrText(pc),a3
1266 moveq #0,d0
1267 bsr MessageWindow
1268 Go_ZeigeSeite
1269 bra ZeigeSeite
1270
1271 FILESTARTRoutine
1272 moveq #0,d2
1273 bra.s FileLocate
1274
1275 FILEENDRoutine
1276 move.l FileSize,d2
1277 subq.l #1,d2
1278
1279 FileLocate
1280 bsr CursorOff
1281 FileLocate2
1282 move.l FileBuffer(Var),a0
1283 cmp.l FileSize,d2
1284 bhs.s NoLocate
1285 move.l d2,d0
1287 divu.w #BytesPerLine,d0
1288 bvs.s NoLocate
1289 clr.w CrY(Var)
1290 swap d0
1291 move.w d0,CrX(Var)

```

Listing: FileMonitor.asm

```

1292 ext.l d0
1293 sub.l d0,d2
1294 lea (a0,d2,1),a0
1295 move.l a0,PageStart(Var)
1296 NoLocate
1297
1298 bra.s Go_ZeigeSeite
1299
1300 STELLERoutine
1301 lea MyStrGadget+si_MaxChars(Mem),a1
1302 move.w #9,(a1)
1303 lea StelleFrage(pc),a0
1304 move.w #ReqX,d2
1305 bsr GetString5
1306 move.w #49,(a1)
1307 tst.b d0
1308 beq.s ReturnSub5
1309 bsr CursorOff
1310 lea StelleAdresse(Mem),a0
1311 moveq #0,d2
1312 GetOffsLoop
1313 move.b (a0)+,d1
1314 beq.s ReadyOffs
1315 cmp.b #'9',d1
1316 bls.s NGreaterDez3
1317 bclr #5,d1
1318 subq.b #'A'-'9'-1,d1
1319 NGreaterDez3
1320 sub.b #'0',d1
1321 bml.s HexStringErr
1322 cmp.b #$F,d1
1323 bhi HexStringErr
1324 lsl.l #4,d2
1325 or.b d1,d2
1326 bra.s GetOffsLoop
1327 ReadyOffs
1328 cmp.l FileSize,d2
1329 blo.s FileLocate2
1330 bsr CursorOn
1331 bra FILEENDRoutine
1332
1333
1334 ** Ganze Datei einlesen
1335 LoadBuffer
1336 move.l MyDosBase(Var),a6
1337 move.l FileBuffer(Var),a0
1338 move.l a0,d2
1339 move.l FileSize,d3
1340 move.l ReadFileHandle(Var),d1
1341 beq.s NotLoad
1342 CALLVEC Read
1343 st.b FileLoaded(Var)
1344 ReturnSub5
1345 NotLoad rts
1346
1347 EditHEXRoutine
1348 clr.b d7
1349 bra.s SetEditMode
1350 EditASCIIRoutine
1351 st.b d7
1352 SetEditMode
1353 tst.b EditOn(Var)
1354 beq.s MustEditMode
1355 cmp.b AsciiOn(Var),d7
1356 beq.s ReturnSub5
1357 MustEditMode
1358 bsr.s EDITMODERoutine
1359 bra.s SetEditMode
1360
1361 # Editmode wechseln
1362 EditmodeRoutine
1363 EDITMODERoutine
1364 tst.b EditOn(Var)
1365 bne.s NStartEdit
1366 EditmodeRoutine2
1367 bsr.s EditStartRoutine
1368 NStartEdit
1369 bsr.s Go_EditCursor2
1370 Not.b AsciiOn(Var)
1371 bra.s Go_EditCursor2
1372
1373 EditAusRoutine
1374 tst.b EditOn(Var)
1375 beq.s ReturnSub5
1376 EditStartRoutine
1377 Not.b EditOn(Var)
1378 Go_EditCursor2
1379 bra EditCursor2
1380
1381 # Dateimemory freigeben
1382 CloseDatei
1383 move.l (_SysBase).w,a6
1384 move.l FileBuffer(Var),d0
1385 beq.s NoFreeBuf
1386 clr.l FileBuffer(Var)
1387 move.l d0,a1
1388 move.l FileSize,d0
1389 add.l #BytesPerPage,d0
1390 CALLVEC FreeMem
1391 NoFreeBuf
1392 sf.b Changed(Var)
1393 sf.b FileLoaded(Var)
1394 clr.l PageStart(Var)
1395 suba.l FileSize,FileSize
1396 rts
1397

```

Listing: FileMonitor.asm



```

1398
1399 ** Neues CLI-Fenster oeffnen
1400 NEWCLIRoutine
1401     move.l   MyDosBase(Var),a6
1402     lea      EmptyStream(pc),a0
1403     move.l   a0,d1
1404     move.l   #MODE_NEWFILE,d2
1405     CALLVEC  Open
1406     move.l   d0,d2
1407     move.l   d0,d3
1408     beq.s    ReturnSubs
1409     lea      NewCliCom(pc),a0
1410     move.l   a0,d1
1411     CALLVEC  Execute      CLI-Befehl starten
1412     move.l   d3,d1
1413     jmp      _LVOClose(a6)
1414
1415 ** Von / In Seitenpuffer kopieren
1416 CopyFromPage
1417     move.l   PageStart(Var),a0
1418     lea      PagePuffer(Mem),a1
1419     bra.s    CopyPage
1420
1421 CopyToPage
1422     move.l   PageStart(Var),a1
1423     lea      PagePuffer(Mem),a0
1424
1425 CopyPage
1426     move.w   #BytesPerPage/4-1,d0
1427 copyPageLoop
1428     move.l   (a0)+,(a1)+
1429     dbf      d0,copyPageLoop
1430     rts
1431
1432 ** Seitenpuffer - Scrollroutinen
1433 PufDown
1434     lea.l    PagePuffer(Mem),a0
1435     move.w   #BytesPerPage-BytesPerLine-1,d0
1436 PufdLoop
1437     move.b   (a0,d0.w),BytesPerLine(a0,d0.w)
1438     dbf      d0,PufdLoop
1439     lea      PagePuffer(Mem),a0
1440     move.l   PageStart(Var),a1
1441     moveq    #BytesPerLine/4-1,d0
1442 Pufdadd

```

Listing: FileMonitor.asm

```

1443     move.l   (a1)+,(a0)+
1444     dbf      d0,Pufdadd
1445     rts
1446
1447 PufUp
1448     lea.l    PagePuffer(Mem),a0
1449     move.w   #BytesPerPage-BytesPerLine/4-1,d0
1450 PufupLoop
1451     move.l   BytesPerLine(a0),(a0)+
1452     dbf      d0,PufupLoop
1453     lea      PagePuffer+BytesPerPage-BytesPerLine(Mem),a0
1454     move.l   PageStart(Var),a1
1455     lea      BytesPerPage-BytesPerLine(a1),a1
1456     moveq    #BytesPerLine/4-1,d0
1457 Pufupadd
1458     move.l   (a1)+,(a0)+
1459     dbf      d0,Pufupadd
1460     rts
1461
1462
1463 * Seitenanzeige neu aufbauen
1464 ZeigeSeiteImmer
1465     bset     #1,OldPageStart+3(Var) *unmoeglicher PageSt
1466     art!
1467 ZeigeSeite
1468     move.l   a2,-(sp)
1469     move.l   PageStart(Var),a2
1470     cmp.l    OldPageStart(Var),a2
1471     beq.s    NZeige
1472     moveq    #TOP,d1
1473     moveq    #Lines-1,d2
1474 SeiteLoop
1475     movem.l  a0/a1/a6/d0-d1,-(sp)
1476     move.l   MyRastPort(Var),a1
1477     move.l   MyGfxBase(Var),a6
1478     moveq    #LEFT,d0
1479     CALLVEC  Move
1480     bsr.s    ConvertLine
1481     movem.l  (sp)+,a0/a1/a6/d0-d1
1482     addq.w   #8,d1
1483     lea      6*4(a1),a1
1484     dbf      d2,SeiteLoop
1485     bsr      NZeige
1486     bsr      CursorOn
1487     move.l   (sp)+,a2
1488     bra      CopyFromPage

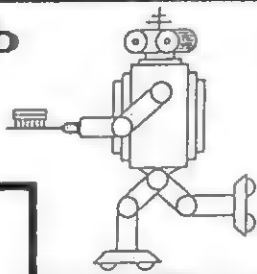
```

Listing: FileMonitor.asm

## Technischer Vertrieb

**Ulrich Wegener**

Obentrautstraße 35  
3000 Hannover 21



### Diskettenlaufwerke

NEC-Laufwerke, durchgeführter Bus	
3,5 intern	165
3,5 extern, abschaltbar	225
5,25 extern, abschaltbar	265

### SCSI-Kontrolller

■ SCSI-Geräte können gleichzeitig betrieben werden. z.B. Fest- und Wechselplatten, optische Laufwerke, Streamer, Laserdrucker etc.  
incl. ALF2-Software, autoboot 645

### SCSI-Festplatten

von 20MB bis 1,2Gb z.B.:	
42MB, Quantum, 19ms	1395
45MB, Fujitsu, 25ms	1265
90MB, Fujitsu, 25ms	1985
135MB, Fujitsu, 20ms	2395
180MB, Fujitsu, 20ms	2765
44MB Wechselplatte, 5,25 25ms	2365

### Software

Pinball Magic	65
Wonderboy In Monsterland	85
It Came From The Desert	95
X-Out	65
Shufflepuck Cafe	75
Rock'n'Roll	75
Hillsfar	85
Day Of The Pharaoh	85
Great Courts	85
Indiana Jones	65
Deluxe Paint III	285
Turbo Print II	95
Superbase II	235
Suprbase profesional	385

### RAM-Erweiterung

0,5MB A500, mit Uhr, abschaltbar	215
1,8MB A500, mit Uhr, abschaltbar	685

A2000 bis 8MB, 2MB bestückt	845
A2000 8MB bestückt	2255

### Turbokarten

68030 - 25 MHz	2695
68030 - 40 MHz	3895
Turbo-PC-Karte incl. Laufw.+Softw.	1065

Videotextdecoder	285
BTX - Decoder MultiTerm, neue Vers.	125

Multisync - Monitore	ab 1355
----------------------	---------

24-Nadeldrucker Citizen Swift	935
-------------------------------	-----

### Modems

300/1200 Baud, Hayes, Terminalprog.	255
300/1200/2400 Baud, Hayes, Terminal	365

### Tip des Monats !

Filecard 20MB, autoboot, abschaltbar formatiert, mit WB1.3 + Backup-Prog.	995
---	-----

**Persönliche Beratung und Bestellannahme**  
Montag bis Freitag von 17 - 21 Uhr

**TEL: 0511 / 32 61 52**

# Listing

```

1487
1488 Pen1          moveq    #1,d0
1489 * aktuellen Farbstift aendern
1490 Pen           movem.l  a0-a2/a6/d1,-(sp)
1491             move.l    MyGfxBase(Var),a6
1492             move.l    MyRastPort(Var),a1
1493             CALLVEC   SetAPen
1494             movem.l    (sp)+,a0-a2/a6/d1
1495             rts
1496
1497
1498 # Zeile in Zeilenpuffer hex/ascii schreiben
1499 * a2=erstes Byte
1500 ConvertLine
1501             movem.l    a0-a1/d0-d1,-(sp)
1502             moveq     #5,d1
1503             lea        Line(Mem),a0
1504             tst.b      FileLoaded(Var)
1505             bne.s      ConvLineLoop
1506             moveq     #77,d0
1507 ClearLine
1508             move.b     #" ",(a0)+
1509             dbf        d0,ClearLine
1510             bra.s      NoConv
1511 ConvLineLoop
1512             move.l     (a2)+,d0
1513             bsr.s      LONGToHEX
1514             addq.l     #8,a0
1515             move.b     #32,(a0)+
1516             dbf        d1,ConvLineLoop
1517             lea        -6*4(a2),a2
1518             moveq     #6*4-1,d1
1519 ConvLineLoop2
1520             move.b     (a2)+,d0
1521             cmp.b      #32,d0
1522             bhs.s      Nopoint
1523             moveq     #" ",d0
1524 Nopoint
1525             move.b     d0,(a0)+
1526             dbf        d1,ConvLineLoop2
1527 NoConv
1528             move.l     MyRastPort(Var),a1
1529             lea        Line(Mem),a0
1530             moveq     #78,d0
1531             CALLVEC   Text
1532             movem.l    (sp)+,a0-a1/d0-d1
1533             rts
1534
1535 *** d0 = INT, a0=Zieladresse *** keine Register veraend
ert ***
1536 LONGToHEX
1537             movem.l    d0-d2/a0,-(sp)
1538             moveq     #32/4-1,d1
1539 ToHEXLoop
1540             rol.l      #4,d0
1541             move.b     d0,d2
1542             lsl.b      #4,d2
1543             lsr.b      #4,d2
1544             move.b     d2,(a0)
1545             cmp.b      #5a,d2
1546             blo.s      Lower10
1547             addq.b     #'A'- '9'-1,(a0)
1548 Lower10
1549             add.b      #'0',(a0)+
1550             dbra       d1,ToHEXLoop
1551             movem.l    (sp)+,d0-d2/a0
1552             rts
1553
1554 # Dateilaenge ermitteln
1555 GetSize
1556             movem.l    d0-d3/a0-a2/a6,-(sp)
1557             move.l     MyDosBase(Var),a6
1558             move.l     ReadFileHandle(Var),d1
1559             moveq     #0,d2
1560             moveq.l     #OFFSET_END,d3
1561             CALLVEC   Seek
1562             move.l     ReadFileHandle(Var),d1
1563             moveq     #0,d2
1564             moveq.l     #OFFSET_BEGINNING,d3
1565             CALLVEC   Seek
1566             move.l     d0,FileSize
1567             movem.l    (sp)+,d0-d3/a0-a2/a6
1568             rts
1569
1570
1571 GetFileName
1572             IFD ARP
1573             lea        RealFileName(Mem),a1
1574             bsr        ShiftTest
1575             bna        GetFileName2
1576             move.l     MyArpBase(Mem),d0
1577             beq        GetFileName2
1578             move.l     d0,a6
1579             move.l     a0,-(sp)
1580             lea        DirName(Mem),a0
1581 CopyDir
1582             move.b     (a1)+,(a0)+
1583             bne.s      CopyDir
1584             lea        DirName(Mem),a0
1585             CALLVEC   BaseName
1586             move.l     d0,a2
1587             lea        FileName(Mem),a1
1588             move.l     a1,a0
1589 CopyName
1590             move.b     (a2)+,(a0)+
1591             bne.s      CopyName
1592             move.l     d0,a2

```

Listing: FileMonitor.asm

```

1592             clr.b      (a2)
1593             lea        FileReq(Mem),a0
1594             move.l     a0,a2
1595             move.l     (sp)+,(a2)+
1596             move.l     a1,(a2)+
1597             lea        DirName(Mem),a1
1598             move.l     a1,(a2)+
1599             move.l     MyWindow(Var),(a2)+
1600             CALLVEC   FileRequest
1601             tst.l      d0
1602             beq.s      ExitGetName
1603             lea        RealFileName(Mem),a2
1604             move.l     a2,a0
1605             lea        DirName(Mem),a1
1606 CopyDirName
1607             move.b     (a1)+,(a2)+
1608             bne.s      CopyDirName
1609             lea        FileName(Mem),a1
1610             CALLVEC   TackOn
1611             moveq     #1,d0             Z-Flag loeschen
1612             ENDC
1613 ExitGetName
1614             rts
1615
1616 # Öffnet ein Fenster, schreibt Textseite in A0, fordert
Eingabe #
1617 GetString5
1618             movem.l    a1-a6/d2-d4,-(sp)
1619             lea        StelleAdresse(Mem),a1
1620             moveq     #2,d0
1621 Go_GetString4
1622             moveq     #1,d1
1623             bra.s      GetString4
1624 GetString3
1625             movem.l    a1-a6/d2-d4,-(sp)
1626             lea        HexSearchString(Mem),a1
1627             moveq     #3,d0
1628             bra.s      Go_GetString4
1629 GetString2
1630             movem.l    a1-a6/d2-d4,-(sp)
1631             moveq     #3,d0
1632             moveq     #0,d1
1633             lea        AsciiSearchString(Mem),a1
1634             bra.s      GetString4
1635
1636
1637 GetFileName2
1638             moveq     #2,d0
1639             moveq     #0,d1
1640             move.w      #ReqX,d2
1641             movem.l    a1-a6/d2-d4,-(sp)
1642 GetString4
1643             move.w      d2,MyRequester+rq_LeftEdge(Mem)
1644             move.w      d0,d2
1645             move.l     a1,MyStrGadget(Mem)
1646             clr.w      MyStrGadget+si_BufferPos(Mem)
1647             clr.w      MyStrGadget+si_DisPos(Mem)
1648             lea        ReqText1(Mem),a5
1649             move.l     a0,it_Itext(a5)
1650             move.l     d1,it_NextText(a5)
1651             beq.s      NSecondText
1652             lea        it_SIZEOF(a5),a1
1653             move.l     a1,it_NextText(a5)
1654             move.l     a0,it_Itext(a1)
1655             clr.l      it_NextText(a1)
1656 GetTNext
1657             tst.b      (a0)+
1658             bne.s      GetTNext
1659             move.l     a0,it_Itext(a5)
1660 NSecondText
1661             lea        MyRequester(Mem),a0
1662             lea        G(a0),a1
1663             move.w      #ReqY,(a1)+
1664             move.l     #ReqWidth<<16+ReqHeight,(a1)+
1665             clr.l      (a1)+
1666             lea        MyGadget1(Mem),a2
1667             move.l     a2,(a1)+
1668             lea        MyStrGadget(Mem),a3
1669             move.l     a3,gg_SpecialInfo(a2)
1670             lea        MyGadget2(Mem),a3
1671             move.l     a3,(a2)
1672             movem.l    a0/a1/a5,-(sp)
1673             lea        GadgetText(pc),a1
1674             lea        MyText1(Mem),a0
1675             moveq     #ReqWidth>>2,d1
1676             lsl.w      #2,d1
1677             divu       d0,d1
1678             move.w      d1,d0
1679             sub.w      #7*8,d0
1680             lsr.w      #1,d0
1681             move.w      d2,d3
1682             moveq     #0,d3
1683             subq.w      #3,d0
1684 GadgetInit
1685             clr.l      (a3)
1686             move.l     a3,-gg_SIZEOF(a3)
1687             addq.l     #4,a3
1688             move.w      d0,(a3)+
1689             add.w      d1,d0
1690             move.w      #ReqHeight-22,(a3)+ gg_TopEdge
1691             move.l     # (7*8+9)<<16+8+9,(a3)+
1692             addq.l     #2,a3
1693             move.l     # (RELVERIFY+ENDGADGET)<<16+REQGADGET:BOOLGAD
GET,(a3)+
1694             lea        MyBorder2(Mem),a5
1695             move.l     a5,(a3)+ gg_GadgetRender

```

Listing: FileMonitor.asm



```

1696 addq.l #4,a3
1697 move.l a0,(a3)+ gg_IntuiText
1698 move.w #0300,(a0) it_FrontPen
1699 move.l #00050005, it_LeftEdge(a0)
1700 move.l a1, it_IText(a0) Textadresse
1701 addq.l #8,a1
1702 lea it_SIZEOF(a0),a0
1703 addq.l #8,a3
1704 move.w d3,(a3)+ gg_GadgetID
1705 addq.l #4,a3
1706 addq.w #1,d3
1707 cmp.w d2,d3
1708 bne.s GadgetInit
1709 lea MyBorder3(Mem),a0
1710 move.l a0,MyGadget2+gg_GadgetRender(Mem)
1711 movem.l (sp)+,a0/a1/a5
1712 lea ReqBorder(Mem),a2
1713 move.l #02020005,bd_FrontPen(a2)
1714 lea XY5(pc),a3
1715 move.l a3,bd_XY(a2)
1716 move.l a2,(a1)+
1717 move.l a5,(a1)+
1718 move.b #1,rq_BackFill(a0)
1719 move.l MyWindow(Var),a1
1720 move.l MyIntBase(Var),a6
1721 CALLVEC Request
1722 tst.b d0
1723 beq.s RequestErr
1724 move.l #REQUEST,d3
1725 bar.s MsgWait
1726 move.l MyIntBase(Var),a6
1727 lea MyGadget1(Mem),a0
1728 move.l MyWindow(Var),a1
1729 lea MyRequester(Mem),a2
1730 CALLVEC ActivateGadget
1731 move.l #GADGETUP,d3
1732 bar.s MsgWait
1733 move.l MyStrGadget(Mem),a0
1734 move.b (a0),d0
1735 beq.s ExitRequest
1736 move.w gg_GadgetID(a2),d0
1737 subq.w #1,d0 -0+ ID
1738 ExitRequest
1739 movem.l (sp)+,a1-a6/d2-d4
1740 rts
1741 RequestErr
1742 lea NoReqText(pc),a3
1743 moveq #0,d0
1744 bar MessageWindow
1745 bra.s ExitRequest
1746
1747
1748
1749 MagWait move.l MyUserPort(Var),a0
1750 move.l (_SysBase).w,a6
1751 CALLVEC WaitPort
1752 move.l MyUserPort(Var),a0
1753 CALLVEC GetMsg
1754 move.l d0,a1
1755 move.l 1m_Class(a1),d2
1756 move.l 1m_IAddress(a1),a2
1757 CALLVEC ReplyMsg
1758 cmp.l d3,d2
1759 bne.s MsgWait
1760 rts
1761
1762
1763 *** Unterroutine fuer HelpWindow und RequesterWindow (
a6=gfx,a2=Win.,a3=Text) ***
1764 Textout2 *** d3=y, wie Textou
t
1765 moveq #5,d0 x
1766 Textout3
1767 move.w d1,-(sp)
1768 move.w d3,d1 y => d1
1769 addq.w #8,d3 y=Y+8
1770 move.l wd_RPort(a2),a1 rastport
1771 CALLVEC Move ...
1772 move.w (sp)+,d0
1773 move.l a3,a0 Textadr fuer Textroutine set
zen
1774 move.l wd_RPort(a2),a1 rastport
1775 CALLVEC Text raus damit
1776 rts
1777
1778 *** Ausgabe von DOS-Fehlermeldungen, a6=dos
1779 IO_Error_Out
1780 movem.l a0-a3/d0-d2,-(sp)
1781 CALLVEC IoErr
1782 lea DosErrorText+IoErrorString-IoErrorText(Mem),
a1
1783 moveq #100,d2
1784 ConvToDec
1785 divu d2,d0
1786 move.b d0,(a1)
1787 add.b #'0',(a1)+
1788 swap d0
1789 ext.l d0
1790 divu #10,d2
1791 bne.s ConvToDec
1792 lea DosErrorText(Mem),a3
1793 moveq #0,d0
1794 bar.s MessageWindow
1795 moveq #0,d1 Wenn, IO_Error_Out aufgerufe
n, dann war Ladefehler
1796 movem.l (sp)+,a0-a3/d0-d2

```

Listing: FileMonitor.asm

```

1797 NoErr
1798 ReturnSub11
1799 rts
1800
1801
1802 LooseChangesTest
1803 move.b Changed(Var),d0
1804 not.b d0
1805 bne.s ReturnSub11
1806 lea WarnungsText(pc),a3
1807 YesNoRequest
1808 movem.l a0-a6/d0-d7,-(sp)
1809 lea MyText1(Mem),a2
1810 move.w #0200,(a2)
1811 move.l #00050003, it_LeftEdge(a2)
1812 lea OkText(pc),a0
1813 move.l a0, it_IText(a2)
1814 moveq #0,d0
1815 bra.s MessageWeiter
1816 * Gibt einen Requester mit Infos fuer den User aus, war
tet auf ReturnTaste *
1817 MessageWindow
1818 movem.l a0-a6/d0-d7,-(sp)
1819 suba.l a2,a2
1820 MessageWeiter
1821 lea ReqText3(Mem),a1
1822 move.l a3, it_IText(a1)
1823 lea -it_SIZEOF(a1),a5
1824 move.l d0, it_NextText(a1)
1825 beq.s NDreiTexte
1826 lea -it_SIZEOF(a5),a0
1827 move.l a5, it_NextText(a1)
1828 move.l a0, it_NextText(a5)
1829 Loop1 tst.b (a3)+
1830 bne.s Loop1
1831 move.l a3, it_IText(a5)
1832 Loop2 tst.b (a3)+
1833 bne.s Loop2
1834 move.l a3, it_IText(a0)
1835 NDreiTexte
1836 lea CancelText(Mem),a3
1837 move.w #0200,(a3)
1838 move.l #00050003, it_LeftEdge(a3)
1839 lea AbbruchText(pc),a0
1840 move.l a0, it_IText(a3)
1841 suba.l a0,a0
1842 moveq #0,d0
1843 move.l $VANILLAKEY,d1
1844 move.w #ReqWidth+ReqWidth/4,d2
1845 moveq #ReqHeight*2/3-4,d3
1846 move.l MyIntBase(Var),a6
1847 CALLVEC AutoRequest
1848 move.l a2,d1
1849 beq.s ExitReq
1850 tst.b d0
1851 ExitReq movem.l (sp)+,a0-a6/d0-d7
1852 rts
1853
1854 * d1=y
1855 DrawHor1
1856 move.l MyGfxBase(Var),a6
1857 move.l MyRastPort(Var),a1
1858 moveq #2,d0
1859 movem.l a1/d1,-(sp)
1860 CALLVEC Move
1861 movem.l (sp)+,a1/d1
1862 move.w #637,d0
1863 jmp _LVODDraw(a6)
1864
1865 # d0=x d1=starty
1866 DrawVert1
1867 move.l MyGfxBase(Var),a6
1868 move.l MyRastPort(Var),a1
1869 movem.l a1/d0/d1,-(sp)
1870 CALLVEC Move
1871 movem.l (sp)+,a1/d0/d1
1872 add.w #10,d1
1873 jmp _LVODDraw(a6)
1874
1875 * d7=MouseX Leiste=a0 LeistenRoutinen=a1 LCount=d0
1876 TestLeiste
1877 move.w (a1)+,d0
1878 cmp.w (a0)+,d7
1879 blo.s ExitL
1880 cmp.w (a0),d7
1881 bhi.s TestLeiste
1882 lea Routines(pc),a1
1883 add.w d0,a1
1884 move.l a1,-(sp)
1885 jsr (a1)
1886 move.l MyDosBase(Var),a6
1887 moveq #8,d1
1888 CALLVEC Delay
1889 bra.s MsgTest
1890 LeisteCall
1891 move.l (sp),a1
1892 jsr (a1) Funktion anspringen
1893 MsgTest move.l (_SysBase).w,a6
1894 move.l MyUserPort(Var),a0
1895 CALLVEC GetMsg
1896 tst.l d0
1897 beq.s LeisteCall
1898 move.l d0,a1
1899 CALLVEC ReplyMsg
1900 move.l (sp)+,a1
1901 ExitL rts

```

Listing: FileMonitor.asm

# Listing

```

1902
1903
1904 # d0=Count d1=Y a0=LeisteEnde a1=LeisteInfoCoords
1905 GetLeiste
1906 moveq #0,d2
1907 bra.s SetLeiste
1908 GetLeisteLoop
1909 addq.w #1,d2
1910 cmp.b #160,-(a0)
1911 bne.s GetLeisteLoop
1912 subq.l #2,a1
1913 SetLeiste
1914 movem.l a0-a1/d0/d1,-(sp)
1915 move.w d2,d0
1916 lsl.w #3,d0
1917 add.w #-639,d0
1918 neg.w d0
1919 move.w d0,(a1)
1920 bsr.s DrawVert1
1921 movem.l (sp)+,a0-a1/d0/d1
1922 subq.w #1,d0
1923 bpl.s GetLeisteLoop
1924 rts
1925
1926 * a0=MenuLeiste, a1=Definitionen
1927 GetMenu
1928 move.l a0,-(sp)
1929 move.l a2,mu_FirstItem(A0)
1930 moveq #0,d1
1931 GetMenuLoop
1932 lea ft_SIZEOF+mi_SIZEOF(a2),a0
1933 move.b 3(a1),d0
1934 bpl.s NextFound
1935 suba.l a0,a0
1936 NextFound
1937 move.l a0,(a2)+ NextItem
1938 clr.w (a2)+ LeftEdge
1939 move.w d1,(a2)+ TopEdge
1940 move.w #MWidth-MLeft,(a2)+ Width
1941 move.w #MHeight,(a2)+ Height
1942 move.w (a1)+(a2)+ Flags
1943 clr.l (a2)+ MutualExcl.
1944 btst.b #0,-1(a1)
1945 beq.s NCheck
1946 move.b (a1),-1(a2)
1947 NCheck lea mi_SIZEOF-mi_ItemFill1(a2),a0
1948 move.l a0,(a2)+ ItemFill1
1949 clr.l (a2)+ SelectFill1
1950 move.b (a1)+(a2)+ Commkey
1951 addq.l #1,a1
1952 addq.l #1,a2
1953 clr.l (a2)+ SubItem
1954 clr.w (a2)+ NextSelect
1955 # IntuiText erstellen
1956 move.w #1<<7+(a2)+ Pen/Paper
1957 clr.w (a2)+ DrawMode/Even
1958 move.w #CHECKWIDTH,(a2)+ LeftEdge
1959 clr.w (a2)+ TopEdge
1960 clr.l (a2)+ Font
1961 move.l a3,(a2)+ String
1962 NextTextLoop2
1963 tst.b (a3)+
1964 bne.s NextTextLoop2
1965 clr.l (a2)+ Next
1966 add.w #MHeight+3,d1
1967 tst.b d0
1968 bmi.s ExitMenuLoop
1969 add.b d0,d1
1970 bra.s GetMenuLoop
1971 ExitMenuLoop
1972 move.l (sp)+,a0
1973 move.l mu_NextMenu(a0),d1
1974 move.l d1,a0
1975 bne.s GetMenu
1976 ExitGetMenu
1977 rts
1978
1979 # a0= Definitionen a1=Strukturen a3=Texte
1980 GetMenuLeisten
1981 moveq #MLeft,d1
1982 moveq #0,d0
1983 GetMenuLeistenLoop
1984 move.b (a0)+,d0
1985 beq.s ExitGetMenuLeisten
1986 lea mu_SIZEOF(a1),a2
1987 move.l a2,(a1)+
1988 move.w d1,(a1)+
1989 add.w d0,d1
1990 clr.w (a1)+
1991 move.w d0,(a1)+
1992 move.l #MHeight<<16+MENUENABLED,(a1)+
1993 move.l a3,(a1)+
1994 NextTextLoop1
1995 tst.b (a3)+
1996 bne.s NextTextLoop1
1997 move.l a2,a1
1998 bra.s GetMenuLeistenLoop
1999 ExitGetMenuLeisten
2000 clr.l -mu_SIZEOF(a2)
2001 rts
2002
2003 Datas
2004 DosName DOSNAME
2005 IntName INTNAME
2006 GfxName GRAFNAME
2007 ArpNm dc.b "arp.library",0

```

Listing: FileMonitor.asm

```

2008 FontName dc.b "topaz.font",0
2009 WindowName dc.b "*** Filemonitor V2.0 *** - "
2010 NoFileText dc.b "Keine Datei geladen",0
2011 ScreenTitle dc.b "FILEMONITOR V2.0 written in 1989
by HOCHISOFT - "
2012 dc.b 169,"1990 DMV-Verlag",0
2013 ConsoleName dc.b "console.device",0
2014 NewCliCom dc.b "NEWCLI >"
2015 EmptyStream dc.b "NIL:",0
2016 even
2017 MyTextAttr dc.l FontName
2018 dc.w #
2019 dc.b FS_NORMAL
2020 dc.b FPF_ROMFONT
2021
2022 XY dc.b -1,-7,8+9-1,-7,8+9-1,2,-1,2,-1,-6
2023 XY2 dc.b -1,-7,8,-7,8,2,-1,2,-1,-6
2024 XY3 dc.b 2,2,7*8+7,2
2025 dc.b 7*8+7,8+6,2,8+6,2,2
2026 XY4 dc.b 0,0,7*8+9,0
2027 dc.b 7*8+9,8+8,0,8+8,0,0
2028 even
2029 XY5 dc.w 2,1,ReqWidth-3,1
2030 dc.w ReqWidth-3,ReqHeight-2,2,ReqHeight
-2
2031 dc.w 2,1
2032
2033
2034 MItem1
2035 ItemCount set 0
2036 MNr set 0
2037 Mitem COMMSEQ,<"L">,0,LOAD
2038 Mitem COMMSEQ,<"S">,0,SAVE
2039 Mitem 0,0,MVY,NEWCLI
2040 Mitem COMMSEQ,<"Q">,LAST,QUIT
2041 ItemCount set 0
2042 Mitem CHECKIT!CHECKED,%10,0,CASESENSEOFF
2043 Mitem CHECKIT,%1,MVY,CASESENSEON
2044 Mitem COMMSEQ,<"F">,0,ASCIIFIND
2045 Mitem COMMSEQ,<"N">,0,ASCIINEXT
2046 Mitem COMMSEQ,<"P">,MVY,ASCIIPREV
2047 Mitem COMMSEQ,<"H">,0,HEXFIND
2048 Mitem COMMSEQ,<"W">,0,HEXNEXT
2049 Mitem COMMSEQ,<"R">,LAST,HEXPREV
2050 ItemCount set 0
2051 Mitem COMMSEQ,<"T">,0,FILESTART
2052 Mitem COMMSEQ,<"B">,MVY,FILEEND
2053 Mitem COMMSEQ,<"G">,LAST,STELLE
2054 ItemCount set 0
2055 Mitem COMMSEQ,<"-">,0>EditAus
2056 Mitem COMMSEQ,<"X">,0>EditHEX
2057 Mitem COMMSEQ,<"A">,MVY>EditASCII
2058 Mitem COMMSEQ,<"U">,LAST,RESTORE
2059
2060 MenuLeisten
2061 MENULEISTE MenuLeiste1
2062 MENULEISTE MenuLeiste2
2063 MENULEISTE MenuLeiste3
2064 MENULEISTE MenuLeiste4
2065 dc.b #
2066 even
2067
2068 Menus
2069 ActualMenu set 0
2070 MENU LOAD
2071 MENU SAVE
2072 MENU NEWCLI
2073 MENU QUIT
2074 ActualMenu set 1
2075 MENU ASCIIFIND
2076 MENU ASCIINEXT
2077 MENU ASCIIPREV
2078 MENU HEXFIND
2079 MENU HEXPREV
2080 MENU HEXNEXT
2081 ActualMenu set 2
2082 MENU FILESTART
2083 MENU FILEEND
2084 MENU STELLE
2085 ActualMenu set 3
2086 MENU EditAus
2087 MENU EditHEX
2088 MENU EditASCII
2089 MENU RESTORE
2090
2091 Commands
2092 COMMAND Pfeil_Hoch,CUp
2093 COMMAND Pfeil_Runter,CDown
2094 COMMAND Pfeil_Links,CLft
2095 COMMAND Pfeil_Rechts,CRght
2096 COMMAND TAB_Taste>Editmode
2097 COMMAND F10_Taste>EditStart
2098
2099 GadgetTexte
2100 OkText dc.b " Ok! ",0
2101 AbbruchText dc.b "Abbruch",0
2102 dc.b "Vorige!",0
2103
2104
2105 ZeigerText
2106 dateizeig dc.b " DateiZeiger: "
2107 dateizeigenda
2108 FilePosText dc.b "00000000 Dateilaenge: "
2109 FileLenText dc.b "00000000"
2110 ZeigerTextEnde
2111

```

Listing: FileMonitor.asm



```

2112 IoErrorText dc.b "AmigaDos Fehler "
2113 IoErrorString ds.b 3
2114 dc.b 0,-2
2115 even
2116 DoubleSpace dc.b " "
2117 Space dc.b " ",0
2118
2119 MenuTexts
2120 TEXT MenuLeiste1,<"Projekt ">
2121 TEXT MenuLeiste2,<"Suchen ">
2122 TEXT MenuLeiste3,<"Position ">
2123 TEXT MenuLeiste4,<"Edieren ">
2124
2125 **** Projekt
2126 TEXT LOAD,<"Laden">
2127 TEXT SAVE,<"Speichern">
2128 TEXT NEWCLI,<"CLI-Fenster oeffnen">
2129 TEXT QUIT,<"Ende">
2130 **** Suchen
2131 TEXT CASESENSOFF,<"Gross/Klein gleich">
2132 TEXT CASESENSON,<"Gross/klein verschieden">
2133 TEXT ASCIIIFIND,<"String suchen">
2134 TEXT ASCIIINEXT,<"naechsten String suchen">
2135 TEXT ASCIIIPREV,<"vorigen String suchen">
2136 TEXT HEXFIND,<"Hexfolge suchen">
2137 TEXT HEXNEXT,<"naechste Hexfolge suchen">
2138 TEXT HEXPREV,<"vorige Hexfolge suchen">
2139 **** Position
2140 TEXT FILESTART,<"Dateianfang">
2141 TEXT FILEEND,<"Dateiende">
2142 TEXT STELLE,<"HEX-Offset eingeben">
2143 **** Edieren
2144 TEXT EditAus,<"Edieren aus">
2145 TEXT EditHEX,<"HEX-Haelfte edieren">
2146 TEXT EditASCII,<"ASCII-Haelfte edieren">
2147 TEXT RESTORE,<"Seiten-Undo">
2148
2149 GadgetLeiste1
2150 LEISTE <"Dateistart ">
2151 LEISTE <"ende ">
2152 LEISTE <"Ed HEX ">
2153 LEISTE <"Ed *** ">
2154 LEISTE <"Ed ASCII ">
2155 LEISTE <"Seite rueckwaerts ">

```

Listing: FileMonitor.asm

```

2156 LEISTE <"Zeile rueckwaerts ">
2157 Leiste1Size equ *-GadgetLeiste1
2158 GadgetLeiste2
2159 LEISTE <"Seite vorwaerts ">
2160 LEISTE <"Zeile vorwaerts ">
2161 Leiste2Size equ *-GadgetLeiste2
2162
2163 even
2164 L1Info
2165 LEISTCOM FILESTART
2166 LEISTCOM FILEEND
2167 LEISTCOM EditHEX
2168 LEISTCOM EditAus
2169 LEISTCOM EditASCII
2170 LEISTCOM PAGEUP
2171 LEISTCOM CUP2
2172 L1InfoEnde
2173 L2Info LEISTCOM PAGEDOWN
2174 LEISTCOM CDown2
2175 L2InfoEnde
2176
2177 LadenFrage dc.b "Welche Datei wollen Sie laden?"
2178 SpeichernFrage dc.b "Unter welchem Namen speichern?"
2179 StelleFrage dc.b "Bei welchem HEXOffset (z.B. 'A2b ')",0
2180 dc.b "positionieren ?","0
2181 HEX_suchText dc.b "Bitte geben Sie die zu suchende ",0
2182 dc.b "Hex-Folge ein (z.B. '2e6C'):",0
2183 HEX_ErrText dc.b "Fehlerhafte HEXZahl!",0
2184 ASCII_suchText dc.b "Welche Zeichenfolge suchen Sie? ",0
2185 LowMemText dc.b "Nicht genugend Speicher fuer P uffer!",0
2186 dc.b "Bitte andere Tasks entfernen",0
2187 dc.b "und wieder versuchen.",0
2188 NoReqText dc.b "Zu wenig Speicher fuer Requeste r!",0
2189 WarnungsText dc.b "Aenderungen verlieren?","0
2190 SaveVerifyText dc.b "Alte Datei wirklich ueberschrei ben?","0

```

Listing: FileMonitor.asm

## OASE



### - Die deutsche Softwarequelle -

- 1- RETURN TO EARTH V1.1. Spannendes Weltraumstrategiespiel mit Action.
- 2- KAMPF UM ERIADOR. Das bekannte Fantasy-Strategiespiel für 2 Spieler.
- 6- LUCKY LOSER. Wirklich gut gelungener Glückspielautomat mit allen Extras.
- 8- TEXTVERARBEITUNG. Einfach bedienbar, z.B. Briefe. Sehr Komfortabel!
- 9- VIDEO DATEI. Bringt endlich Ordnung im Cassettenchaos (LP-Datei auf 10).
- 12- HAUSHALTSBUCH. frei definierbare Konten. Sehr gute Dokumentation.
- 13- MOUNTAIN CAD. Das professionelle CAD-Grafik-Paket. TopHit!
- 14- WIZARD OF SOUND. Perfektes Musikprogramm + viele Instrumente.
- 16- VIRUS STOP. Bekämpft die gängigsten Viren (auch fiese Linkviren!).
- 21- STAR TREK SPIEL. Das Superstrategiespiel mit toller Grafik+Sound (1MB!).
- 24- ETIKETTEN. Bedruckt Ihre Etiketten. Mit Titelgrafiken (selbst erstellbar!)
- 26- GIROMAN. Verwaltet Ihr komplettes Girokonto. Einfache Steuerung.
- 30- MORIA. Super-Abenteuerrollenspiel für Fantasy Fans (1MB!).
- 31- BATTLEFORCE. Strategischer Kampf der Titanen. TopHit! Sehr Komplex.
- 33- PETERS QUEST. Friedliches Hüpf- und Sammelspiel. Joysticksteuerung.
- 35- BILLARD. Ausgezeichnetes Billardspiel für ein oder zwei Spieler.
- 41- DISKETTENMONITOR. Sehr umfangreiche Möglichkeiten. Erstklassig!
- 42- MANDELBROT deluxe. Erstellt farbenprächtige Grafiken. Mit Einsteigerkurs!
- 44- ASTRONOMIE. Sehr komplexes Sternennprogramm mit vielen Daten.
- 47- ATLANTIS. Grandioses, spannendes Fantasyspiel (1MB!). Gute Grafik.
- 48- SCHACH. Spielstarkes Schachspiel mit guter Grafik. Maussteuerung.
- 50- LABELPAINT. Ausgezeichnetes Etikettenbeschriftungsprogramm mit Grafik!

Alle Programme der deutschen OASE-Serie werden auf hochwertigen 2DD-Wolf-Qualitätsdisketten fehler- und virusfrei kopiert und haben eine **DEUTSCHE ANLEITUNG!** je Disk **DM 10,-**

### -Deutsche Anwendersoftware-

- 100- FAKTURA deluxe  
Diese Komplettversion enthält eine Fakturierung mit Rechnungen, o.P., Mahnungen, Kundendaten, Lagerlisten, etc. Als Bonbon kann der Rechnungskopf durch ein IFF-Bild gestaltet werden! Das Programm gibt es mit deutscher Anleitung nur bei uns (benötigt 1MB). **DM 30,-**
- 101- FIBU deluxe  
Universelle Finanzbuchhaltung mit frei definierbaren Konten, Bilanzen, G+V-Rechnung, etc. Sehr umfangreich. *Mandantenfähig!* Ideal für Klein- und Mittelstandsunternehmen. Diese Profiversion erhalten Sie mit deutscher Anleitung nur bei uns (benötigt 1MB). **DM 30,-**

### -102- TeX-SCHRIFTSATZ

TeX ist ein professionelles Programm (**KEINE Demo!**) zum Setzen von Texten. Mit seiner Funktionsvielfalt und der Möglichkeit der Grafikbindung ist es ideal für das Schreiben von Dokumenten, Urkunden, Büchern oder sogar Doktorarbeiten. Wir bieten TeX in einer Grundversion mit deutscher Anleitung und NEC P6 Druckertreiber (nur Draft!) an für nur (benötigt 1MB!) **DM 40,-**  
Epson 9-Nadeltreiber+Fonts DM 60,-/ Epson 24-Nadeltreiber+Fonts DM 30,-

### -103- BIORHYTHMUS deluxe

Dieses Programm erstellt Ihren ganz persönlichen Biorhythmus mit umfangreichen Auswertungen. Werte lassen sich sogar ausdrucken. **DM 20,-**

### -104- QUIZ

In diesem interessanten Fragespiel mit über 500 Fragen können bis zu 4 Spieler Ihr Wissen testen. Die Fragen (oft grafisch nett untermauert!) kommen dabei aus allen Bereichen (z.B. Film, Politik, Geographie, Geschichte, Technik, etc.) Ein tolles Programm für die ganze Familie! Maussteuerung! (1 MB!) **DM 20,-**

### -105- SUPERDAT deluxe

Sehr leicht bedienbare Dateiverwaltung für Adressen, Mitgliederlisten, Lagerhaltung, etc. Frei definierbare Datenmaske, daher sehr variabel einsetzbar. Umfangreiche Datenabfragen und Druckfunktionen sind möglich. Komplette Benutzerführung in deutsch. **DM 30,-**

### -106- GNUTeX

Funktionsgraphengenerator für professionelle Kurvenerzeugung. Direkter Ausdruck und Einbindung in TeX möglich. Mit 30-seitiger Anleitung (1MB). **DM 40,-**

### -107- CONTENTS

Komplettes Disketteninformationssystem mit ausgezeichnetem Diskettenverwaltung, direktem Diskettenzugriff und Programmfunktionen. Endlich haben Sie einen sehr guten Überblick über Ihre Softwaresammlung. Natürlich mit komfortablem Etikettendruck der Inhaltsverzeichnisse. **DM 39,-**

### -108- DUNGEON FLIPPER

Ein rasanter Flipper mit vielen Extras. Bis zu 4 Spieler können Ihr Können auf den 2 Spielebenen beweisen. Tolle Grafik, digitaler Sound! (benötigt 1MB) **DM 39,-**

### -109- EINKOMMENSTEUER '89

Dieses Programm erstellt auf sehr einfache Weise (voll Menügesteuert!) Ihre komplette Lohn- und Einkommensteuererklärung. Deckt fast 100% aller Normal- und Sonderfälle ab. Mit jährlichem Updateservice! **DM 59,-**

Versandkosten (Porto+Verpackung)

Inland: V-Scheck DM 3,- / Nachnahme DM 7,-  
Ausland: V-Scheck DM 6,- / Nachnahme DM 15,-



**Computertechnik**  
Inh.: Rainer Wolf

Deipe Stegge 187  
4420 Coesfeld

**Tel.: 02541/2874**  
**Fax: 02541/71172**



**Liebe Leser,**  
in dieser Rubrik finden Sie Rat und Hilfestellung zu Ihren kleineren und größeren Programmierproblemen und Informationen zu den in der AMIGA DOS vorgestellten Programmen und Produkten. Wir sind natürlich jederzeit bemüht, die eingehenden Leserfragen zu beantworten. Doch haben Sie bitte Verständnis, daß wir nicht alle eingehenden Briefe persönlich beantworten können. Oft erreichen uns mehrere Briefe zum gleichen Thema, einer davon wird dann stellvertretend für alle in unserer Zeitschrift beantwortet.

**Dringliche Probleme, die das Heft betreffen, lassen sich womöglich besser telefonisch regeln. Rufen Sie dienstags unsere Hotline an. Von 17-20h stehen wir Ihnen mit Rat und Tat zur Seite, wenn Sie eine der folgenden Nummern wählen:**

**05651/809-740 (bis 744)**

**Ihre AMIGA-DOS-Redaktion**

# SIE FRAGEN, WIR ANTWORTEN!

## Probleme mit Fish 187 - DiskPerf

Ich habe einige Probleme mit dem PD-Programm DiskPerf auf der Fish-Disk 187. Das Programm dient dazu, die Datenübertragungs-Geschwindigkeit einer Festplatte zu messen. Wenn ich das Programm mit dem Aufruf "Diskperf DH0:" starte, führt es ein Directory-Scan und einen Seek-Read-Test durch. Danach erscheint die Fehlermeldung "Could not get 524288 Bytes of Memory". Obwohl 1 MByte zur Verfügung steht, mangelt es angeblich an Speicherplatz.

Meine Frage: Fehlen beim Start des Programms irgendwelche Parameter, oder muß ich noch andere Dinge beachten?

M. Sihn, Mühlacker

Das Problem liegt offensichtlich daran, daß Ihrem Rechner nicht genügend Speicher zur Verfügung steht, zumindest nicht genug "durchgängiger Speicher". Wahrscheinlich verwenden Sie noch eine alte DiskPerf-Version. Es sind bereits einige Updates zu diesem Programm erschienen, auf die Sie zurückgreifen sollten. Diverse PD-Vertreiber werden Ihnen sicherlich weiterhelfen können.

(Red.)

## Adresse gesucht

Auf der Messe AMIGA '89 wurde ein Umbausatz für die 500er-Modelle vorgestellt, der aus einem Gehäuse besteht, in dem die alte Platine nur eingeschraubt wird und durch ein Kabel mit Erweiterungssteckplätzen verbunden wird. Für Laufwerke sollen wie beim 2000er Laufwerkschächte vorhanden sein. Das Gehäuse ist mit einem stärkeren Netzteil versehen, damit die Spannungsversorgung für Erweiterungen gewährleistet ist. Leider weiß ich nicht, welche Firma diesen Bausatz anbietet. Könnt Ihr mir die Adresse mitteilen?

C. Stelter, Wilhelmshaven

Das System heißt MW 550 und wird von der Firma Computer & Zubehör Miky Wennigatz angeboten. Die vollständige Adresse lautet:

Computer & Zubehör  
Miky Wennigatz  
Jägerweg 31  
8031 Gilching  
Tel.: 08105/24540  
Fax: 08105/24530

## Info zum BSG9-Virus

Der Virus nimmt grundsätzlich den ersten Befehl, der in der Startup-Sequenz vor-

kommt, an. Ein Beispiel: Steht als erstes der Befehl Set-Patch, so kopiert der BSG9-Virus diesen Befehl in das Devs-Directory (jedoch ohne Namen aber mit Datums- und Größenangabe). Danach stimmt er den Namen Set-Patch an und kopiert sich ins C-Verzeichnis. Nach Ausführung des Befehls in der Startup-Sequence, kopiert er sich ins RAM und leitet den Befehl in das Verzeichnis Dev um. Wird nun auf Diskette etwas abgespeichert, so überprüft BSG9 den ersten Eintrag in der Startup-Sequence. Ist dieser Eintrag zufällig AddBuffer, kopiert er diesen Befehl ohne Namen ins Devs-Directory und sich selbst unter dem Namen AddBuffer ins C-Verzeichnis. Ich hoffe, die Leser können mit dieser Information etwas anfangen.

D. Kewitsch, Hamburg

## Schreib/Lese-Fehler

Seit rund fünf Monaten bin ich stolzer Besitzer eines Amiga 500. Seit geraumer Zeit bereiten mir die Anzeigen von Read/Write Errors bei mehreren Disketten doch erhebliches Kopfzerbrechen. Zunächst schob ich diese Probleme auf die Lagerung der Disketten in der Nähe meiner Amateurfunkstation, bzw. in der Nähe des Netzteils. Jedoch stellte ich nach einiger

Zeit des Testens fest, daß es hieran nicht liegen kann. Wodurch werden diese Diskettenfehler verursacht? Gibt es ein sicher arbeitendes und leicht zu bedienendes Programm, um solche Disketten zu reparieren?

H. Lange, Horb 1

Zu diesem Problem erreichten uns bereits mehrere Leseranfragen, die alle das gleiche Problem schilderten. Stellvertretend für diese Anfragen haben wir den Leserbrief von Herrn Lange veröffentlicht.

Da Sie dieses Problem bei mehreren Disketten festgestellt haben, deutet es auf einen Link-Virus hin. Hier tut sich besonders der Lamer Extremator hervor, der - einmal auf Diskette vorhanden - diese gänzlich unleserlich macht. Genauer gesagt, auf der vom Lamer befallenen Diskette wird ein Datenblock mit dem Wort "Lamer!" beschrieben und damit gänzlich vom Virus verändert, so daß ein Read/Write Error die Folge ist.

Um sich vor solchen Virenangriffen schützen zu können, gehört ein gutes Virenschutzprogramm zur Grundausstattung eines jeden Amiga. Wir können Ihnen aus dem Bereich der Public Domain das Programm VirusX 4.0 empfehlen, das bei allen versierten PD-Anbietern im Angebot



ist. Sie können sich das Programm in Ihr C-Verzeichnis der Arbeitsworkbench kopieren und in der Startup-Sequenz festhalten. Danach erscheint am oberen Bildschirmrand ein kleines Fenster, das Sie jederzeit aktivieren können, um eine neu eingelegte Diskette zu überprüfen. Das Programm erkennt eigenständig die geläufigsten Viren und kann bei Erkennen die verseuchte Diskette auch reparieren.

Um herauszufinden, wann sich ein Linkvirus in Ihrem System "breitgemacht" hat, können Sie auch unser Listing LinkChecker in dieser Ausgabe benutzen.

(Red.)

## Probleme beim Seka Assembler

In der AMIGA DOS 2'90 wurde unter dem Titel "Fehler im 68000" auf Seite 112 auf ein merkwürdiges Phänomen im Zusammenhang mit dem Befehl "AND.L A1,D1" verwiesen. Zur Lösung des Problems:

Der Fehler liegt beim Assembler. Der Seka-Assembler mag zwar den Befehl codieren und später auch decodieren, die Befehls-Kombination ist aber im 68000er nicht zulässig. Der Seka steht damit aber nicht allein: auch der eine oder andere versiertere Assembler scheint vor solchen Reinfällen nicht gefeit. Daher auch ein kleines Lob an den PD-Assembler A68K (zum Beispiel Fish 186), der sich standhaft weigerte, diesen Befehl zu übersetzen und dabei folgende Meldung von sich gab: "Addressing mode not allowed here".

G. Steffens, Harsewinkel

## Digi View Gold

In Ihrer Ausgabe vom Februar 1990 haben Sie das Programm Digi View Gold getestet. Da ich dieses Programm gerne erstehen würde, möchte ich Sie bitten, mir eine Adresse mitzuteilen, wo ich dieses Programm bekommen kann.

J. Ehrnsberger, Neumarkt  
Das Programm Digi View Gold finden Sie normalerweise im gut sortierten Fachhandel. Da Sie mit dieser Aussage sicherlich nicht sehr viel an-

fangen können, hier einige Adressen. Digi View Gold ist zu beziehen bei:

GTI GmbH  
Zimmersmühlenweg 73  
6370 Oberursel  
Tel: 06171/73048/9  
Fax: 06171/8302

UNLIMITED  
M. Hottenbacher  
Kehrstr. 23  
6200 Wiesbaden  
Tel: 06121/543848

Creative Computers GmbH  
Lutterothstr. 58  
2000 Hamburg 20  
Tel: 040/407332  
Fax: 040/4919237

Atlantis GmbH  
Dunantstr. 53  
Postfach 1141  
5030 Hürth  
Tel: 02233/41081  
Fax: 02233/46266

amigaOberland  
Hohenwaldstr. 26  
6374 Steinbach  
Tel: 06171/71846  
Fax: 06171/74805

(Red.)

## Anschlußprobleme?

Seit einiger Zeit besitze ich neben dem C64 einen Amiga 500. Daß ich mit diesem neuen Gerät auch gern einen Drucker betreiben würde, versteht sich natürlich von selbst. Ich besitze jedoch einen STAR-LC-10-Color mit serielltem Kabel. Meine Frage dazu:

Kann ich diesen Drucker durch irgendeine Hardware etc. für meinen Amiga nutzen? Kann man den C64 eventuell als Interface benutzen? Welche Möglichkeiten gibt es dazu?

F. Heine, Wolfsburg

Verschiedene Firmen bieten Interfaces an, die Ihr Problem lösen dürften. Um den C64 jedoch als "Druckerpuffer" zu benutzen, bedarf es etwas mehr Hardwarepraxis.

Eine mögliche Konstellation sähe so aus: Verbindung des Amiga-Parallel-Port mit dem Userport des C64, der durch ein spezielles Programm (am besten in Assembler geschrieben, um Timing-Problemen aus dem Weg zu gehen) den Userport als parallelen Port verwaltet. Der RAM-Bereich könnte die empfangenen Daten auffangen und an den Drucker weitergeben. Ver-

schiedentlich wurden in anderen Publikationen schon Bauanleitungen gegeben, einige Hardware-Bücher zum C64 enthalten ebenfalls Hinweise, hier ist allerdings ein Umschauen in der Elektronikbranche vonnöten.

(Red.)

## Termine abfragen

Ich suche nach einer effektiven Möglichkeit, wichtige persönliche Daten und Termine beim Einschalten des Rechners abfragen zu können oder an diese erinnert zu werden. Gibt es eine Batch-Datei für die Startup-Sequenz, oder gibt es ein kurzes Basic-Listing dafür?

B. Haidner, Wien

Sicher ist es möglich, sich ein Programm zu schreiben, das beim Einschalten des Rechners an wichtige Termine erinnern soll. Dazu bedarf es nur der Abfrage der (sofern vorhanden) batteriegepufferten Echtzeituhr im Amiga.

Wir können jedoch auf ein kommerzielles Produkt hinweisen, das wir im Rahmen der Aktuellseiten in Ausgabe 2/90 vorgestellt haben. Das Programm heißt Wer!Was! Wann!Wo!, dessen Demoverision frei weitergegeben werden darf.

(Red.)

## Erweiterungen für Amiga 1000

Ich habe einen Amiga 1000, PAL-Version mit einer Erweiterung von 1MByte (3State). Dazu habe ich einige Fragen: Kann ich beim Amiga 1000 einen Resetbutton einbauen, der auch das Kickstart raus-schmeißt? Wenn ja, wie funktioniert das?

Wie sieht es mit Erweiterungskarten aus? Kann ich beim 1000er auch Turbokarten, PC-Karten und ähnliches einbauen, oder bin ich da auf spezielle Erweiterungen, die es nur für den Amiga 1000 gibt, angewiesen?

T. Hüsemann, Oberhausen

Rein theoretisch müßte ein Hardwarereset, der über den Expansionsport ausgeführt wird, auch den Kickstart löschen. Betreffs der Erweiterung für den Amiga 1000: PC- und Turbokarten können nicht in den Amiga 1000 eingebaut werden.

Erweiterungen werden in der Regel an den Expansionsport angeschlossen. Um am Amiga 1000 auch MS-DOS benutzen zu können, muß ein Sidecar (Spezielle PC-Erweiterung für den Amiga 1000) angeschlossen werden.

(Red.)

## DTP bzw. Musik mit dem Amiga

Ich möchte zwei Vereinszei-tungen herausbringen, was mit den heutigen Computer-möglichkeiten eigentlich kein Problem sein dürfte. Das Zaub-erwort heißt hier Desktop Publishing. Da ich weder Krösus noch Computereffreak bin, ist guter Rat teuer. Was benötige ich unbedingt an Hardware, damit ich überhaupt mit DTP arbeiten kann, und welche Software können Sie, wenn man das Preis/Leistungsverhältnis mit in Betracht zieht, empfehlen? Das zweite Thema, das mir unter den Nägeln brennt, ist Musik. Ich möchte in Verbindung mit meinem Keyboard Casio HT 3000 und der entsprechenden Software die notwendigen Grundlagen schaffen, selbst ein wenig zu komponieren und arrangieren. Ich habe zwar nicht vor, Dieter Bohlen zu ersetzen, aber eigene Musik zu kreieren, ist auch ganz schön. Können Sie mir bezüglich der zu beschaffenden Soft- oder auch Hardware weiterhelfen?

H.-G. Winkler, Markdorf

DTP ist eine faszinierende Möglichkeit, den Computer sinnvoll einzusetzen. Beim Amiga müssen dabei folgende Punkte beachtet werden: Professionelle DTP-Programme arbeiten aus Gründen der Übersichtlichkeit im hochauflösenden Modus (Interlace). Der Standard-Monitor ist für solche Arbeiten kaum geeignet. Inzwischen gibt es aber auch recht preiswerte Komplet-Pakete wie den Publisher (Test in der AMIGA DOS 03/90, Preis: zirka 300,- DM), die mit dem normalen Monitor arbeiten. Mit diesem Programm zum Beispiel lassen sich recht ansehnliche Ergebnisse erzielen. Bleibt das Problem des Druckers: Ein 24-Nadel-Matrix-Drucker gibt zwar zufriedenstellende, aber nicht optimale Ergebnisse ab. Hilfe würde das Programm PixelScript bringen (Test in diesem Heft), welches von Lay-

out-Programmen erzeugte PostScript-Dateien für Matrix-Drucker umwandelt und einen wesentlich besseren Ausdruck ermöglicht. Alles in allem kostet der Einsatz von DTP, wenn er zumindest semiprofessionell sein soll, nicht gerade wenig.

Die Lösung Ihres zweiten Problems heißt MIDI (zu deutsch: Digitale Schnittstelle für Musikinstrumente). Falls Ihr Keyboard über eine MIDI-Schnittstelle verfügt, benötigen Sie ein MIDI-Interface, das den Datenaustausch zwischen Keyboard und Computer ermöglicht. Ferner benötigen Sie noch ein MIDI-fähiges Musikprogramm (beispielsweise Music-Studio von Activision oder das in der letzten Ausgabe vorgestellte Quest I). Ist diese Konstellation gegeben, können Sie Ihren Kompositionsfähigkeiten freien Lauf lassen. Im Musikbuch von DATA Becker ist übrigens eine Bastelanleitung für ein MIDI-Interface enthalten.

(Red.)

### Wie rüste ich einen Amiga 1000 auf?

Wir benutzen einen Amiga 1000 und sind soweit mit Hard- und Software zufrieden. Das einzige Manko besteht in der langen Bootzeit und dem leider nicht ausreichenden Speicherplatz. In den meisten Fachzeitschriften wird immer wieder zu einer Festplatte geraten. Nun meine Frage:

Wie können wir dem 1000er Beine machen, ihn kostengünstig erweitern und welche Festplatte paßt zu ihm bzw. wo bekommt man sie?

B. Haidner, Wien

Die FSE- und CombiTec-Auto-boot-Festplatten werden unter anderem auch für den Amiga 1000 angeboten. Die Kapazität der Platten liegt bei 20 bis 122 MByte.

Die Adressen:

CombiTec Computer GmbH  
Liegnitzer Str. 6-6a  
D-5810 Witten  
Tel.: (02302) 88072

Frank Strauß Elektronik  
Schmiedstr. 11  
D-6750 Kaiserslautern  
Tel.: (0631) 67096-98

### Stichwort "Pointer"

Ich möchte über Genlock einen Pfeil (wie auf der Amiga Workbench vorhanden) in ein laufendes Videobild einblenden. Leider ist im Amiga-Programm außer dem Pfeil noch eine Umrandung und eine Kopfleiste vorhanden. Kann man diese herauslösen, bzw. nur einzelne Files von der Workbench kopieren, um dies zu erreichen, oder ist Ihnen ein anderes Programm bekannt, das nur einen Pfeil (Pointer) auf den Bildschirm bringt?

G. Held, Kutenholz

Falls der von Ihnen gewünschte Pfeil unbeweglich sein kann und nur an unterschiedlichen Stellen auftauchen soll, bietet sich die Möglichkeit an, den Pfeil mit einem Malprogramm zu erstellen und als IFF-Brush (Pinsel) abzuspeichern. Beim Einlesen des Genlock-Bildes können beide Screens aufeinander gelegt werden, wobei der Pfeil an die jeweilige Stelle gesetzt wird und alles zusammen auf das Videoband abgespeichert werden kann. Soll es sich um einen bewegten Zeiger handeln, muß ein Programm geschrieben werden, welches den neuen Pfeil anstelle des Mauspointers legt.

(Red.)

### Amiga 500 und C64

Ich besitze einen Amiga 500 mit Speichererweiterung und gepufferter Uhr (Systemspeicher 1,5 MByte). Außerdem sind eine Festplatte (A590 20 MByte), ein Zweitlaufwerk (3,5 Zoll), ein Drittlaufwerk (5,25 Zoll) und ein Farbdrucker MPS 1500 angeschlossen. Durch Ihren Artikel "BASIC in allen Gassen" hat sich mir die Frage gestellt, ob es irgendwie möglich ist, die Software eines C64-Computers auf dem Amiga zu benutzen. Ich praktiziere dies bereits mit PC-Software (MS-DOS) mit Hilfe eines PC-Emulators (softwaremäßig) einigermaßen erfolgreich.

Gibt es einen C64-Emulator für den Amiga 500, können Sie mir Bezugsadressen nennen oder existieren hardwaremäßige Lösungen?

M. Kottsiepe, Kaarst-Vorst

Gleich zwei C64-Emulatoren waren noch vor gar nicht so langer Zeit auf dem Markt, ei-

ner von ihnen, der 'GO64', arbeitete mit den Programmen des C64 auch zufriedenstellend. Leider besitzen wir im Augenblick keine Informationen, ob diese Emulatoren noch auf dem Markt zu finden sind. Falls Ihnen, liebe Leser, noch etwas dazu einfällt, schreiben Sie uns bitte, wir werden die Infos in einer der nächsten Ausgaben veröffentlichen.

### Betrifft: Programm GREGOR

Wenn ich die Erklärung zu Ihrem Programm GREGOR richtig verstanden habe, dann müßte das Jahr 1600 ein Schaltjahr sein. Nach der Eingabe von 29.2.1600 erhielt ich aber die Meldung "...gibt es nicht". Beim Suchen nach der Ursache glaube ich folgenden Fehler entdeckt zu haben.

Das Programm erklärt zwar in Zeile 15 jedes vierhundertste Jahr zum Schaltjahr, macht das aber in Zeile 22 wieder zunichte (jedes hundertste Jahr ist ein Nicht-Schaltjahr). Eine mögliche Abhilfe wäre die folgende Verlängerung der Zeile 22:

```
22: IF j/100 = INT(j/100) AND  
j/400 <> INT(j/400) THEN
```

Bei mir funktioniert es jedenfalls.

A. Pistor, Wien

### Probleme mit der DATABOX

Als Neuling, der im Besitz eines Amiga 500 ist, habe ich ein paar grundlegende Fragen zur DATABOX:

- 1) Mache ich von der DATABOX eine Kopie, verlangt diese nach dem Start immer das Original. Warum?
- 2) Wie starte ich BASIC ohne Workbench?
- 3) Wie kopiere ich ein Programm der DATABOX auf eine andere Diskette?

Peter Stiller  
1000 Berlin 2

Höchstwahrscheinlich haben Sie Ihre DATABOX-Disketten durch Anklicken auf der Workbench kopiert. Dies ist schon richtig, jedoch müssen Sie die neuen Disketten mit den Original-Namen DATABOX-1 und DATABOX-2 benennen, das 'COPY OF ...' also entfernen. Die Programme werden von den logischen Laufwerken DATABOX-1: und DATABOX-2: gestartet, bei einer 'COPY OF...'-Version wird dieses logische Device nicht mehr erkannt. Wie Sie BASIC ohne Workbench starten, lesen Sie in diesem Heft im Amiga-BASIC-Kurs unter der Überschrift 'BASIC-Diskette die Zweite'. DATABOX-Programme lassen sich aufgrund der fehlenden Icons nur von der Shell

## Die Problemecke

In der Problemecke werden Leserbriefe veröffentlicht, zu denen die Redaktion im Augenblick keine Antwort parat hat. Hier sind Sie, liebe Leser der AMIGA DOS gefragt. Sollten Sie zu einem oder anderen Thema eine Antwort wissen, schreiben Sie uns unter dem Stichwort 'Problemecke'.

### Die PC-Karte im Blickpunkt

Die PC-Kartenbeschleuniger einiger Firmen sind mir schlichtweg zu teuer. In diesem Zusammenhang habe ich einige Fragen an Ihre Redaktion.

- Kann ich die Schaltung, die in der DOS 1'88, S. 48 erschienen ist, auf der PC-Karte 2088 zur Beschleunigung von 4,77 MHz auf 8 MHz verwenden?
- An welchen Custom-Chip und Pin muß ich die Schaltung anschließen?
- Welche Änderungen muß ich sonst an der PC-Karte vornehmen?
- Wo bekomme ich die neueste Version der JANUS-Software, die schon seit einem Jahr angekündigt wurde, her?

G. Bottke, Nürnberg



(CLI) aus kopieren. Dazu nehmen Sie den COPY-Befehl: COPY DATABOX-1: < Programmname > TO RAM: kopiert ein DATABOX-Programm von der Diskette ins RAM, COPY ?

ruft den COPY-Befehl auf, ohne ihn auszuführen, danach eingeben:

RAM: < Programmname > TO DFO:

kopiert auf eine danach eingelegte Diskette (sinnvoll, wenn nur ein Laufwerk vorhanden ist).

Ansonsten kann man mit COPY von jedem Laufwerk zu jedem Laufwerk kopieren.

(Red.)

## Pascal-Ecke in der AMIGA DOS

Mit Interesse las ich Ihren Artikel in der Ausgabe 3/90 über KickPascal auf dem Amiga. Ich bin seit Ende Dezember '89 Besitzer dieses Compilers und trotz der von Ihnen aufgeführten Negativmerkmale sehr zufrieden. Das Ansprechen des Amiga-Betriebssystems ist gut gelungen und die Arbeitsumgebung sehr komfortabel. Zudem ist der Preis recht günstig. Deshalb würde mich eine Pascal-Ecke in Ihrer Zeitschrift besonders interessieren, und ich hoffe auf eine starke Resonanz seitens anderer Leser.

R. Ortner, Marl

Wir haben diesen Brief stellvertretend für alle diejenigen abgedruckt, die mehr über KickPascal in der AMIGA DOS wissen wollen. Unsere Aktion geht jedoch noch weiter: Schreiben Sie uns Ihre Meinung zu dieser 'neuen' Programmiersprache, wir sind weiter gespannt.

(Red.)

## Datums- und Zeitangabe auf dem A500

Ich besitze seit einiger Zeit einen Amiga 500. Im Oktober letzten Jahres kaufte ich mir eine Speichererweiterung A 501 RAM von Commodore. Seither gab mein Amiga beim Starten der Workbench immer die aktuelle Datums- und Zeitangabe aus. Seit einigen Wochen nun läßt die Workbench zwar normal, aber anstatt der Datums- und Zeitansage erhalte ich den Ausdruck [un-

set] [unset] [unset].. Anfänglich dachte ich an eine Verseuchung durch Viren, aber eine Überprüfung mit der Original-Workbench brachte das gleiche Ergebnis. Könnte es eventuell sein, daß meine Speichererweiterung defekt ist? Die Startup-Sequence wurde von mir nicht geändert.

U. Mayer, Rothenburg o.T.

Ihre Speichererweiterung ist nicht defekt, das Problem liegt lediglich darin, daß Ihre batteriegepufferte Uhr neu gestellt werden muß. Stellen Sie dazu in den Preferences die Uhrzeit neu ein, und klicken Sie danach in diesem Menü den Save-Button. Dadurch wird die Zeit wieder auf den aktuellen Stand gebracht und gespeichert.

(Red.)

## XT-Karte und deutscher Zeichensatz

Seit kurzem bin ich Besitzer eines Amiga 2000 mit PC-XT-Karte. Und genau dort liegt mein Problem. Nachdem ich MS-DOS geladen habe, ist leider nur die amerikanische Tastatur aktiv. Ich habe zwar gelesen, daß man in den Dateien CONFIG.SYS und AUTOEXEC.BAT entsprechende Änderungen vornehmen kann, aber damit habe ich keinen richtigen Erfolg.

Deshalb möchte ich Sie bitten, mir doch den vollständigen Inhalt der beiden Dateien zu schreiben, um meinen Amiga auf deutsche Tastatur und einen Epsondrucker einzustellen. Nach dem Umschalten auf die deutsche Tastatur (im Direktmodus) konnte ich den Schrägstrich "\ " auf keiner Taste mehr finden. Ist das ebenfalls korrekt?

F.Zawadski, Rastatt

Um den deutschen Zeichensatz auf der XT-Seite des Amiga zu aktivieren, sollten Sie die AUTOEXEC.BAT um die Zeile KEYB GR erweitern. Ferner tragen Sie dann in der CONFIG.SYS für COUNTRY die Zahl 049 ein. Den Backslash (\) erreichen Sie übrigens mit der Taste über dem Tabulator (links neben der 1). Der Drucker dürfte nun auch keine Schwierigkeiten mehr machen.

(Red.)

## Installations-schwierigkeiten

Vor kurzem legte ich mir die PD-Disketten Fish-Disk Nr. 67 und 158 zu. Als Anfänger auf dem Amiga stellte sich die Installation der Programme DiskX (Fish 158) und Wheel (Fish 67) für mich als sehr schwer heraus. Nachdem ich zum Beispiel DiskX geladen hatte, erscheint im CLI, daß diese Version ARP V33 benötigt.

Unter dem Icon DiskX.Note steht: "Copy AmigaLibDisk 158:Libs/ARP.Library Libs:". Nachdem ich dieses durchführen wollte, meldete mir das CLI, daß die Datei nicht vorhanden ist.

Bei dem Programm Wheel verstehe ich bei der Installation nur Bahnhof...

T. Rubbert, Hamburg

Bei dem Programm DiskX auf der Fish 158 handelt es sich um die Version 2.1, die die ARP-Library benötigt. Bei dieser Library handelt es sich um einen CLI-Ersatz, der sich beispielsweise auf den PD-Disketten Panorama 27c und RPD 122 befindet. Besorgen Sie sich diese ARP-Library, und kopieren Sie sie in das Libs-Verzeichnis der Fish-Disk 158. DiskX sollte dann ohne Probleme laufen.

Bei dem Programm Wheels handelt es sich um ein Spiel, das auf den AmigaBasic-Interpreter zurückgreift. Zu diesem Behufe sollten sie das AmigaBasic auf die Fish-Disk 67 kopieren. Auch dieses Programm sollte nun keinerlei Probleme mehr darstellen.

(Red.)

## Fehler im Etiketten-Printer

Ich mache mir hin und wieder die Mühe und tippe ein interessant erscheinendes Listing ab, wenn es nicht zu lang ist. Und das, obwohl ich der Sprache BASIC kaum mächtig bin. Hier fiel mir aber auf, daß das Programm nur laden und speichern kann, wenn im gleichen Verzeichnis ein Directory namens 'DISKS' aufsucht, welches man sich selbst erstellen muß (zum Beispiel 'MAKEDIR SYS:DISKS'). Die Druckroutine entspricht nicht meinen Erwartungen an ein Normaletikett für 3,5-Zoll-Disketten im Format 70 x 70 mm. Es wäre doch schön, wenn in den ersten beiden

Zeilen für die Rückseite der Disk folgender Text eingebracht werden könnte:

Enable - >

Protect - >

Offensichtlich hat der Autor Etiketten im Format 50 x 70 mm (oder ähnliches) - also nur für die Vorderseite - verwendet. Dementsprechend habe ich mit meinem EPSON-kompatiblen Drucker Schwierigkeiten mit überflüssigen Linefeeds. Eine Bitte daher: Gebt doch mit dem Hinweis auf das anzulegende Verzeichnis auch die Anregung weiter, die Druckroutine so umzuschreiben, daß auch das von mir benutzte Format unterstützt wird.

Ich würde mich freuen, in einer der nächsten Ausgaben etwas in dieser Richtung zu erfahren.

Wilhelm Ditsche  
Buchholz 2

Beim Etiketten-Printer kam es tatsächlich zu einem Abbruch des Programms, wenn das Directory 'DISKS' nicht gefunden wurde. Dieses Verzeichnis steht in Zeile 490 und kann nach Belieben verändert werden. Im Text wurde dies jedoch leider nicht erwähnt. Zum Ausdruck ist zu sagen, daß uns hier nur eine begrenzte Anzahl Drucker zum Austesten zur Verfügung steht. Der eine oder andere Leser wohl deshalb irgendwann einmal Schwierigkeiten bekommen, wenn sein Drucker es mit der EPSON-Kompatibilität nicht so genau nimmt. In dem Fall bleibt es dem geplagten Druckerbesitzer nicht erspart, sich an die Arbeit zu begeben und das Drucker-Handbuch zu wälzen.

Leider hat uns Herr Ditsche nicht mitgeteilt, welchen Drucker er besitzt, in einem solchen Fall wäre es sinnvoll, Drucker-Probleme in die Problemecke zu setzen. Sollte jemand von Ihnen, liebe Leser(innen), das ETI-Programm so verändert haben, daß es wie von Herrn Ditsche gewünscht funktioniert, so würden wir uns freuen, wenn Sie uns schreiben.

(Red.)

★ Bitte beachten Sie folgendes:

Bitte schreiben Sie bei Leserbriefen Ihre vollständige Adresse nicht nur auf den Umschlag, sondern auch auf den Brief; nur so ist eine schnelle Beantwortung möglich. Bei Briefen, die in der Leserbrief-Fubrik abgedruckt werden, behält sich die Redaktion vor, Leserzuschriften in gekürzter Form wiederzugeben.

(Red.)

# Ein toller Dreh

## Kubistisches

Von allen Faszinationen gehört die Faszination der Dimensionen sicherlich zu den spannendsten. Wir, die wir glauben, ein Leben in vier Dimensionen zu verbringen (nämlich in den drei Dimensionen des Raumes und in der der Zeit), sind beispielsweise nicht in der Lage, uns in mehr- oder minderzahlige Dimensionen hineinzudenken.

**S**tellen Sie sich beispielsweise einmal ein Wesen vor, das in der fünften Dimension (was immer das auch sein mag) existiert oder eines, das mit zweien auskommt. Bis auf ein paar begnadete Genies, die behaupten, es zu begreifen, wird die Mehrzahl doch an dieser Vorstellung scheitern. Das ist wenig verwunderlich, da die Zahl derjenigen, die Probleme mit der räumlichen Vorstellung haben, gar nicht einmal so gering ist.

Wir wollen uns in dieser Tüftelecke einmal in die drei Dimensionen eines Würfels begeben und unsere Vorstellungskraft zum Rotieren bringen.

### Die Aufgabe

Wir stellen uns einen Würfel vor, der seinerseits aus 343 kleineren Würfeln besteht

(also je sieben Würfel lang, breit und hoch ist).

Um eine bessere Orientierung innerhalb dieses Gebäudes zu bekommen, geben wir den einzelnen Würfeln Kennungen in Form von drei Zahlen, die die Koordinaten der kleineren Würfel im Raum darstellen. Der Würfel in der unteren linken vorderen Ecke wird demnach 1-1-1 heißen, der in der oberen rechten hinteren Ecke 7-7-7. Sollte sich im Verlauf unserer Reise der Würfel zu drehen beginnen, dann erhält der neue untere linke vordere Teilwürfel die Kennung 1-1-1; die anderen Teile werden entsprechend neu benannt.

Wir betreten nun den Würfel mit der Nummer 1-1-1. In der unteren Box wird ein Weg be-

schrieben, den wir durch das Innere des geometrischen Aufbaus schreiten wollen, wobei der Würfel die schlechte Eigenschaft hat, sich derweilen um eine seiner Achsen drehen zu müssen (siehe Zeichnung).

Unsere Frage:

- In welchem Würfel befinden wir uns am Ende des Weges?

Die Lösung der Frage schreiben Sie bitte auf eine Postkarte und schicken diese an den:

**DMV-Verlag**  
**Redaktion AMIGA DOS**  
**Stichwort "Kubus"**  
**Postfach 250**  
**3440 Eschwege**

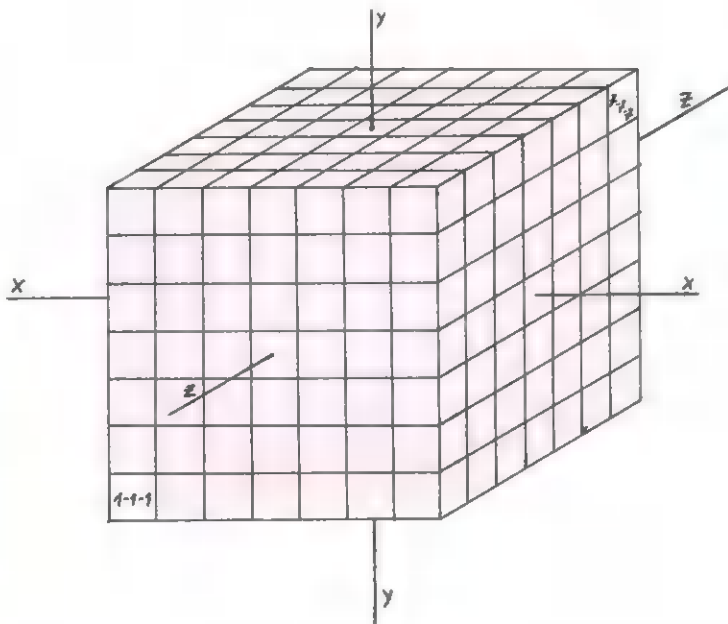
Der Rechtsweg ist ausgeschlossen. Mitarbeiter des DMV-Verlages und deren Angehörige dürfen leider nicht mitmachen. Der Einsendeschluß ist der

15. Mai 1990.

Es gilt das Datum des Poststempels.

### Was gibt es zu gewinnen?

Unter allen richtigen Einsendungen verlosen wir diesmal 10 Exemplare des Spiels X-Out von Rainbow Arts.



### Der Weg:

- voran: 3 Schritte
- rechts: 4 Schritte
- hoch: 2 Schritte
- zurück: 2 Schritte

Ein Ruck geht durch den Würfel, und dieser dreht

sich an der z-Achse um 90 Grad links herum.

- voran: 1 Schritt
- links: 3 Schritte
- hinab: 3 Schritte

Und wieder ruckt es im Gebälk: Der Würfel dreht sich

an der x-Achse 90 Grad nach hinten.

- hinab: 4 Schritte
- rechts: 3 Schritte

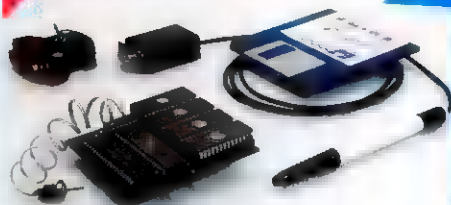
Ein letztes Mal erzittert das Gebilde, und der Würfel dreht sich erneut an der z-

Achse um 90 Grad nach links. Danach verharrt alles in der Stille, und ein letztes Echo der Rotation verhallt in den Tiefen des Würfels.

Wo aber befinden wir uns jetzt?

(jf)

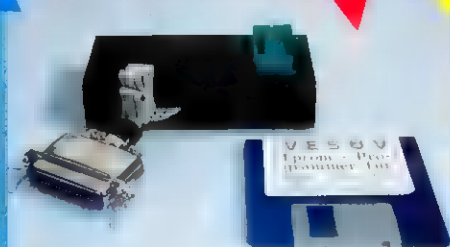




**VIRUS-FALLE** 29,95 DM  
verhindert das Ausbreiten von Boot-Viren.

**LIGHTPEN** ohne Maustasten 79 DM  
**KICKSTART 3** 59,95 DM

Umschaltplatine für 3 verschiedene Kickstarts  
+ 2x original Kickstart-Roms und 1x in Eproms  
Roms/Eproms nicht im Lieferumfang enthalten



**VESUV-AMIGA-Eprommer** 199 DM

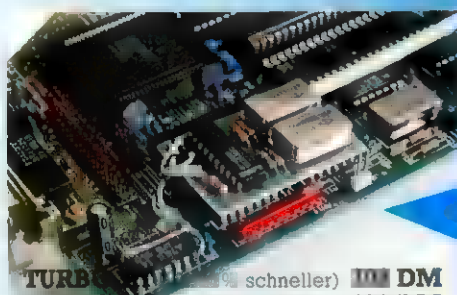
läuft auf A500, 1000 und 2000 + brennt auch 1 MBit-Eproms + „HAPPY“ 3/89 Test-Gesamteurteil „SEHR GUT“ + programmiert die Eproms 2716-27512, 27010 und 27011



**MEGA-DRIVE** 89 DM

1 MByte Diskettenlaufwerk für AMIGA-Dos! + 1,52 MByte unter Amiga-DOS + arbeitet auch mit Ihren alten 880 k Disketten + abschaltbar + durchgeschleifter Bus

**10 HD-Disketten** (1,4 MB) 29,95 DM

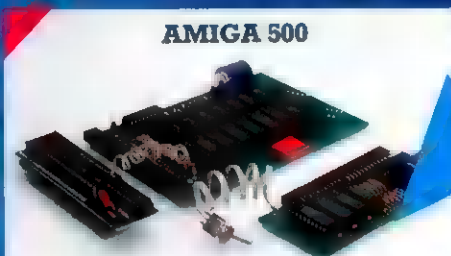


**TURBO RAM** 199 DM  
+ 256 k  
+ schneller (ca. dopp. so schnell)

erweitert Ihre XT-Karte ON BOARD auf 768 KByte! + AT/XT-Karte nicht im Lieferumfang

**Power PC-Board** 798 DM

MACHT IHREN AMIGA 500 IBM-KOMPATIBEL  
+ echte 16-Bit CPU V30 bei einer Taktfrequenz von 8 MHz + Phoenix-Bios mit 768 k Ram unter MS-DOS (im Lieferumfang enthalten) + alle Amiga Ein- und Ausgänge werden unterstützt Maus, Joystick, interne und externe (auch S.25 Zoll) Laufwerke, parallel/ser. Schnittstelle + Festplatten + Superschnelle Bildschirmausgabe: unterstützt Hercules und Farbgrafik! + 1 MByte Ram und Uhr ON BOARD, auch für den Amiga 500 ansprechbar + komplett mit MS-DOS 4.01, GW-BASIC, SHELL, 1MB Ram, Uhr, dt. Handbücher



**AMIGA 500**

**A 512** 179 DM

512 k Speichererweit. + abschaltbar + Uhr

**A2MB/500** 598 DM

1 MByte Ramkarte + mit FAT-AGNUS 1,8 MByte mit BIG-AGNUS volle 2 MByte (Chipram/Fastram) + WELTNEUHEIT: arbeitet mit dem BIG- und dem FAT-AGNUS!



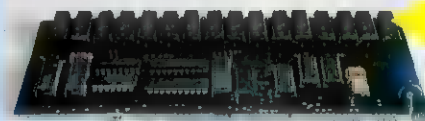
**AMIGA 1000**



**A8MB/1000** 798 DM

8MByte Ramkarte; mit 2MByte bestückt + einfachster Einbau + kein Löten - nur einstecken + abschaltbar + ohne Waitstates

**AMIGA 2000**



**A8MB/2000** 698 DM

8 MByte Ramkarte mit 2 MByte bestückt + zukunftsicher durch 4-MBit-Technologie + auto-konfigurierend + 0-Waitstates + abschaltbar + Anschluß für Reset-Taster

**Multiboard Ramkarte 2MB** 898 DM

mit 4 MByte bestückt 1398 DM

mit 1 MByte bestückt 1998 DM

incl. 4-fach Kickstart-Umschaltung!

...bei uns nutzen Sie heute  
Technologie von morgen

Bitte fordern Sie unseren  
Gratiskatalog an!

**BRANDAKTUELL**

386-si Power Board 1498 DM  
macht aus Ihrer XT-Karte einen 386-SX-Computer + 16 MHz Taktfrequenz und 16 KByte CACHE-Speicher für höchste Geschwindigkeit: macht die XT-Karte ca. 12x schneller! + Steckplatz für 387-SX-Coprozessor.

**HOTLINE**

Technische Fragestunde:  
Mo.-Fr. von 16-17 Uhr. Hier können  
Sie die Entwickler unserer Amiga-  
Produkte sprechen.

022 25/20 61-20 62-20 63

**BIG AGNUS**  
mit Einbauanleitung  
für A 500/2000

**BEI UNS NUR  
99 DM!**

**COMPUTER**  
**ROB MÖLLER**  
TECHNIK

Neuer Markt 21

5309 Meckenheim

Telefon 0 22 25/20 61-6





## Mit Deluxe Paint wie ein Profi arbeiten

Durch Menüs haben wir uns gekämpft, in Icons geschwelgt, Requester aufgerufen und Parameter gesetzt. Wir haben in den bisherigen vier Werkstattbeiträgen das elementare Arbeiten mit Deluxe Paint und seinen unterschiedlichen Funktionen kennengelernt. Die letzte Routine erlangen Sie jedoch nicht durch endloses Studium des Handbuchs, sondern durch angewandte Praxis.

Nach längerem Arbeiten mit Deluxe Paint erarbeitet der Anwender sich viele kleine Tricks, die den Umgang mit Grafik und Deluxe Paint erleichtern.

**I**n diesem letzten Teil wollen wir uns eben diesen kleinen Praxistips zuwenden. Nicht in jedem Falle müssen diese Tips unmittelbar die Arbeit mit Deluxe Paint betreffen. So leistungsfähig das Programm auch ist, in manchen Bereichen, die den Computergrafi-

ker brennend interessieren, muß auch Deluxe Paint passen. Darum werden wir in diesem Teil der Werkstatt nicht Deluxe Paint an sich, sondern noch andere Software ansprechen, deren Leistungen für Deluxe-Paint-Benutzer interessant sind. Wer viel mit Deluxe Paint ar-

Kursfahrplan		
<b>Teil 1:</b> Das Basismenü und seine Funktionen		
<b>Teil 2:</b> Die Kopfleiste und die dort enthaltenen Menüs		
<b>Teil 3:</b> Die Anwendung der in der Kopfleiste enthaltenen Funktionen		
<b>Teil 4:</b> Animation mit dPaint III		
<b>Teil 5:</b> Arbeitstips vom Profi		

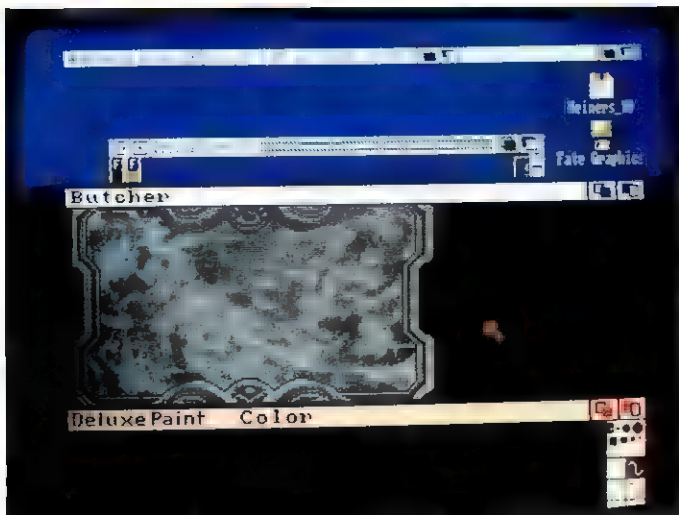
beitet, wird mit der Zeit feststellen, daß die Bedienung des Programmes über die Menüleiste zwar recht komfortabel ist, richtiges, schnelles Arbeiten damit aber etwas problematisch wird.

### Rationelles, schnelles Arbeiten

Werden die gewünschten Kommandos über die Tastatur aufgerufen (fast jeder Deluxe-Paint-Befehl hat ein Tastaturäquivalent), stehen die gewünschten Funktionen

fast augenblicklich zur Verfügung. Natürlich läßt sich nun einwenden, daß hier doch einiges an Tastaturcodes auswendig zu lernen ist, bis Deluxe Paint komplett von hier aus bedient werden kann. Allerdings sollten Sie bedenken, daß sich aus der Fülle der Deluxe-Paint-Funktionen einige herauskristallisieren, die viel öfter benutzt werden als andere. Auch wenn nur diese wenigen verstärkt benötigten Befehle über die Tastatur aktiviert, die wesentlichen anderen Optionen aber wie bisher





Deluxe Paint und Butcher parallel im Einsatz

durch die Menüs bedient werden, ist schon eine deutliche Zeitersparnis zu verzeichnen. Sie sollten auf jeden Fall versuchen, sich ein Blatt mit den einzelnen Tastaturcodes neben den Computer an die Wand zu hängen und jedesmal, wenn eine neue Deluxe-Paint-Funktion aktiviert wird, nachsehen, welche Taste dafür steht. Auf diese Weise lernen Sie diese Codes recht schnell, und ehe Sie sich versehen, kennen Sie die meisten und können Deluxe Paint weit komfortabler als bisher bedienen.

## Dateien hin, Dateien her

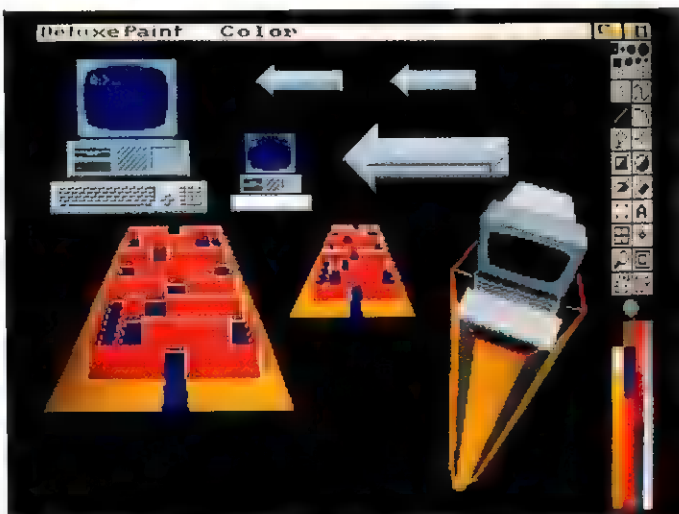
Wenn länger an einem besonders aufwendigen Motiv gearbeitet wird, fallen im Laufe der Arbeit immer mehr Bild- und Brushdateien an, die an-

gelegt wurden, um Zwischenschritte zu sichern. Schnell hat man auf diese Art und Weise ein Verzeichnis gut gefüllt. Deluxe Paint lädt jedesmal, wenn ein <Load>-Requester geöffnet wird, alle Directory-Einträge des Verzeichnisses ein, auf das der Requester gerade eingestellt ist. Sind hier nun viele lange Einträge, resultieren daraus lange Ladezeiten. Dem kann vorgebeugt werden, wenn zum einen versucht wird, den Dateinamen so kurz wie möglich zu halten, zum anderen mit mehreren verschiedenen Verzeichnissen gearbeitet wird, und so die Länge der Directories beschränkt wird. Natürlich bezieht sich dies im wesentlichen auf die Arbeit mit Diskettenlaufwerken, Festplatten haben in der Regel eine genügend kurze Zugriffszeit auf die Daten, so daß Lesevorgänge auch bei wirk-

lich umfangreichen Directories in einem zeitlich vertretbaren Rahmen bleiben. Noch eine Warnung vor zu langen Dateinamen und Leerzeichen in Dateinamen. Tatsächlich ist es schon passiert, daß Verzeichnisse, die randvoll mit überlangen Dateinamen waren, sich aus unerfindlichen Gründen nicht mehr laden ließen. Benutzen Sie Leerzeichen in Dateinamen und wollen diese Dateien später mit der Shell oder dem CLI bearbeiten, müssen Sie darauf gefaßt sein, daß sich auch hier vermeidbare Probleme einstellen. Auch die Benutzung der RAM-Disk ist empfehlenswert, wenn um schnelles Zwischenspeichern geht. Die RAM-Disk braucht etwa 1.5 Sekunden um ein 32-Farben-Bild im HighRes-Modus einzuladen. Allerdings sollten Sie nicht vergessen, die RAM-Disk nach wichtigen Bildern zu durchsuchen, bevor Sie einen Reset ausführen oder den Computer abschalten.

## Das Bildarchiv, des Grafikers Bibliothek

Wird der Computer des öfteren zum Malen benutzt, fallen mit der Zeit eine Vielzahl von Bildern an, ganz zu schweigen von den Brushes, die als Zwischenschritte während der Ausarbeitung des fertigen Bildes anfallen. All diese Grafiken können nun entweder auf Diskette vergammeln oder aber zu einem Archiv zusammengestellt werden, das bei späteren Arbeiten herangezogen werden kann. Ein solches Archiv erfordert natürlich ein gewisses Maß an Organisation. Am besten ist es, nach Abschluß jedes Projektes die vorhandenen Motive zu sichten. Kleinere Brushes können auf einem Bildschirm zusammengestellt und abgespeichert werden. Dies geht natürlich nur, wenn alle Brushes mit denselben Farben arbeiten. Wird eine Brush aus einer anderen Palette hier hineingesetzt, übernimmt sie natürlich die aktuell eingestellten Farbwerte und sieht dann entsprechend poppig bunt aus. Beginnt das Archiv zu wachsen, stellt sich bald die Frage nach der Organisation. Die Frage "Wie finde ich in meinem Archiv etwas?" gewinnt an Bedeutung. Sollten Sie einen Drucker besitzen, ist es eine gute Idee, jeden kompletten Bildschirm,



Solcherart können auf einem Archivbild viele verschiedene Brushes platzsparend zusammengestellt werden

## Sie besitzen einen AMIGA?

Sie wollen ihn erweitern?

Sie wissen nicht wie?

**Aber wir !!!**

**Wir bieten für AMIGA:**

- Speichererweiterungen
- Festplatten
- Wechselfestplatten
- Tapestreamer
- Turboprozessorkarten
- AMIGA-Reparatur in eigener Werkstatt

Auch für Ihren DOS-Rechner oder Bridgecard bieten wir Zubehör oder Komplettlösungen.

Rufen Sie an oder kommen Sie vorbei:

# ABAG

Andreas Bergmann & Adalbert Geyer GdB

Schönauer Str. 17b  
Tel. 06 21/78 74 01

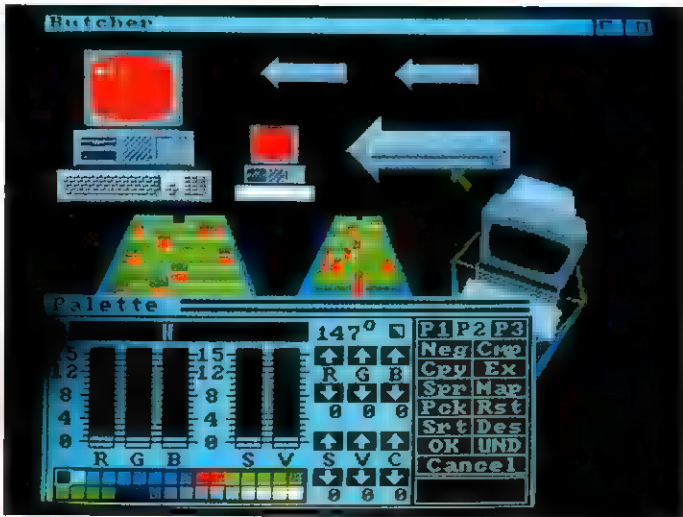
6800 Mannheim 31  
Fax: 06 21/78 74 09

## CSV-HIGHLIGHTS

Commodore		
Amiga Enhancer-Paket (Kickstart V 1.3)		38,-
Commodore Farbmonitor 1084		599,-
Commodore Amiga 500		889,-
Speichererweiterung auf 1 MB mit Uhr		1799,-
Commodore Amiga 2000		1799,-
Amiga 2000 + Farbmonitor 1084		2369,-
3.5" Laufwerk intern für Amiga 2000		199,-
PCXT-Karte mit 5 1/4" Laufwerk		699,-
AT-Karte mit 5 1/4" Laufwerk		1999,-
SCSI Controller Commodore 2090 A		643,-
20 MB-Festplatte für Amiga 2000 mit SCSI Controller Comm. 2090 A (autobootend)		979,-
40 MB-Festplatte mit Controller 2090 A		1399,-
20 MB-Filecard (Seagate, 40 ms) für A 2000 mit PC-Karte oder A 2090		649,-
30 MB-Filecard (Seagate, 40 ms)		779,-
40 MB-Filecard (Western Digital, 29 ms)		879,-
50 MB-Filecard (Seagate, 40 ms)		1049,-
2 MB-RAM Erweiterungskarte für A 2000		949,-
aufrüstbar bis 8 MB (Commodore A 2090)		229,-
Externes 3.5" Laufwerk abschaltbar		899,-
Externe A 500 Festplatte 20 MB Commodore		
Atari		
Festplatte Atari Megaflo 30		879,-
Festplatte Atari Megaflo 60		1369,-
1040 STFM + Monochrommonitor SM 124		1189,-
1040 STFM + SM 124 + Megaflo 30		2039,-
Atari ST + Monochrommonitor SM 124		1499,-
Atari Computer Mega ST 1 mit Maus + Monochrommonitor SM 124		1499,-
Mega ST 1 + SM 124 + Megaflo 30 MB		2349,-
Atari Mega ST 2 + Monochrommonitor SM 124		2199,-
Atari Mega ST 2 + SM 124 + Megaflo 30		3049,-
Atari Mega ST 4 + Monochrommonitor SM 124		3199,-
Atari Mega ST 4 + Monochrommonitor SM 124 + Festpl. Megaflo 30		4049,-
Supercharger für Atari ST		749,-
Epsondrucker (dt. Handbücher)		
Anschlußfertig an AMIGA, Schneider PC, Atari ST und Mega, sonstige IBM-Kompatibel		
Atari ST und Mega, sonstige IBM-Kompatibel		419,-
LO 400 (24-Nadeldrucker)		689,-
LO 550 (24-Nadeldrucker)		889,-
LO 850 (24-Nadeldrucker)		1369,-
Tintenstrahl drucker IX 800 (9 Düsen, NLO, max. 240 Zeichen/Sekunde)		569,-
Stardrucker (dt. Handbücher)		
LC-10 mit Centronicsinterface		429,-
LC-10 Color Farbdruker mit Centronics		569,-
LC 24-10 mit Centronicsinterface		649,-
NEC-Drucker (dt. Handbücher)		
NEC P 6 Plus 1299,-		449,-
NEC P 7 Plus 1749,-		249,-
Speichererw. für A 1000 (Amiga 1050)		69,-
Druckerkabel 5 m lang für Amiga, ST		29,-
NEC Farbmonitor Multisynch 3 D		1499,-
Targa Multisynch (1028 mm, 1024x768)		1099,-

Versandkostenpauschale: Inland DM 12,-, Ausland DM 40,- je Paket. Lieferung nur gegen NN oder Vorauskasse. Ausland nur Vorauskasse. Preise gültig ab 12.3.90.

**CSV RIEGERT GmbH**  
Gärtnersstr. 4, 7320 Göppingen  
Tel. 07161/3591, Fax 07161/13587



**Kontrolle über Farben und Register – Butcher ist die ideale Deluxe-Paint-Ergänzung**

den Sie mit Bildteilen gefüllt haben, einmal auszudrucken, den genauen Dateinamen und den Namen der Diskette darauf zu notieren und die Ausdrücke zu sammeln.

Wird dieses Konzept konsequent verfolgt, können die Drucke in einem Ordner abgelegt werden, und fortan suchen Sie Ihre Bilder nicht mehr auf Diskette, sondern in Ihrem Archivordner.

Festplatten zeichnen sich zwar durch einen schnellen Datenzugriff aus, von daher ist das Durchsuchen diverser Dateien nach einem bestimmten Motiv weniger zeitraubend als mit einer Diskettenstation. Trotzdem empfiehlt sich auch hier ein Archiv, und sei es aus Gründen der Übersichtlichkeit.

## Interessantes über IFF

Als sich die Programmierer von Electronic Arts bei der Arbeit an Deluxe Paint der Problematik widmeten, Grafiken auf Diskette zu speichern, stellte sich die Frage, in welchem Format dies geschehen soll. Die Lösung war das IFF-Format. Hier wurden einige wichtige Vorgaben erfüllt, die eine weitere Verarbeitung der Bilder ermöglichen.

IFF – das steht für Interchange File Format, die Dateistruktur eines IFF-Bildes wird von fast jedem anderen Programm erkannt. Eine IFF-Datei kann aber nicht nur Grafikdaten enthalten. Das IFF-Format ist durchaus auch in der Lage, Texte, Sounddaten und Soundsamples abzuspeichern. Rein theoretisch kann

eine IFF-Datei sogar gleichzeitig Grafikdaten, Sounddaten und Textdaten enthalten.

## Butcher – Der Grafikmetzger

Schon bald wird der Computergrafiker feststellen, daß Deluxe Paint ihm zwar die Kontrolle über die Farbe verleiht, ein weitergehender Zugriff auf die Farbreister allerdings verwehrt bleibt. Insbesondere in diesem Bereich eröffnet das Programm Butcher von Eagle Tree Software völlig neue Arbeitsperspektiven.

Stellen Sie sich bei der Arbeit mit Deluxe Paint folgenden Problemfall vor. Sie haben mit der Spritzpistole gearbeitet und auf einem fast fertigen Bild eine Weile zwei verschiedene Farben versprüht. Plötzlich stellen Sie fest, daß die beiden zuletzt benutzten

Zeichenfarben eigentlich hätten vertauscht werden müssen. Ginge es nun um zwei große Flächen, könnten wir das Problem von Deluxe Paint aus mit zwei Füllaktionen beheben. Aber hier geht es um viele hundert verschiedene Pixel, die keine zusammenhängende Fläche ausmachen. Die einzige Methode, das Problem schnell zu lösen, wäre, einfach die Farbwerte in der Palette neu zu regeln. Wurde die Farbe außerdem noch in anderen Motivteilen verwendet, verändern wir deren Farbwerte natürlich auch, aber das ist dann eine Anzeige mit der schnellen Lösung. Wirklich fix behoben ist der Sorgenfall mit dem Butcher. Er versetzt den Anwender in die Lage, Farbreister zu vertauschen, womit unser Problem behoben wäre. Aber der Butcher hat noch andere Vorzüge. Ist ausreichend Spei-

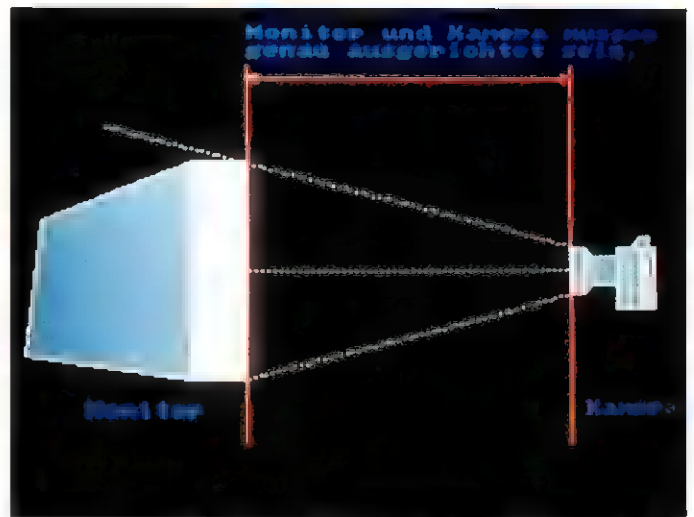
cher vorhanden, können Butcher und Deluxe Paint gleichzeitig betrieben werden. Beide Programme unterstützen, falls vorhanden, RAM-Disks, so daß sich der Datentransfer zwischen den beiden Tasks zeitsparend über dieses Speichermedium abwickeln läßt. Ohne RAM-Disk müssen die Bilder auf Diskette gespeichert, und vom anderen Programm wiederum von dort gelesen werden.

Der Butcher kann Bitplanes und Farben separieren, er arbeitet mit drei frei wählbaren, unterschiedlichen Paletten gleichzeitig, er sortiert die Pixel eines Bildes nach Farbe und Menge und kann außerdem noch vieles mehr. Tatsächlich hat sich der Butcher 2.0 in der Praxis als ideale Ergänzung zu Deluxe Paint herauskristallisiert. In diesem Zusammenhang sei er auch anderen Deluxe-Paint-Benutzern ans Herz gelegt.

## Deluxe Paint und Video

Eines der beliebtesten Peripheriegeräte, das auf den Wunschzetteln von Computergrafikern ganz oben steht, ist ein Videodigitizer. Scheinbar das ideale Bilderfassungsgerät, um Vorlagen vom Papier in den Computer zu übertragen.

Den Grafiker wird dabei wohl weniger am eigentlichen Vorgang des Digitalisierens interessiert sein. Sein Thema wird die nachträgliche Modifikation der Bilder sein. Dabei stellt sich gleich zu Anfang ein elementares Problem. Wirklich gute Ergebnisse



**Ausrichtung einer Kamera zum Monitor bei Bildschirmfotos**



**Der AMIGA-DOS-Animator ermöglicht Animationen auch mit Deluxe Paint II**



werden nur erzielt, wenn mit möglichst vielen Farben digitalisiert wird. Wenn weniger Farben gewählt sollten, dann sollte das Motiv einen möglichst guten Kontrast haben. Andererseits kann aber selbst Deluxe Paint III (bei Deluxe Paint II nicht vorhanden) nur maximal im <Halbbricht>-Modus arbeiten, und der stellt, wie Sie sich sicher erinnern können, 64 Farben dar.

Um die Bilder mit Deluxe Paint zu verarbeiten, muß beim Digitalisierungsvorgang die Farbenanzahl auf 32 bzw. 64 eingestellt werden.

Einige Digitalisierungs-Softwarepakete legen die Digigrafik nicht im IFF-Format ab, so daß das Bild nicht ohne weiteres in Deluxe Paint eingeleiten werden kann. Hier schafft das Programm Butcher Abhilfe. Als eines der besten Werkzeuge, deren Leistungsumfang über das von Deluxe Paint hinausgeht, werden wir uns diesem Programm später noch ausführlich widmen. An dieser Stelle interessiert uns jedoch nur die Fähigkeit des Butchers, die unterschiedlichsten Grafikdateiformate einlesen zu können. Im Programm kann dann der Bildschirmmodus geändert werden, und das Motiv wird auf das neue Format und dessen Farbanzahl konvertiert. Die Grafik kann nach den Veränderungen als IFF-Datei abgespeichert werden, die ohne Probleme in Deluxe Paint eingeleiten werden kann.

## Video und Animation

Wer dem letzten Teil unserer Werkstatt aufmerksam gefolgt

ist, wird festgestellt haben, daß insbesondere die Erstellung der vielen verschiedenen Bilder sehr zeitaufwendig ist. Das Digitalisieren von Bildvorlagen kann auch hier Zeit sparen helfen. Natürlich können Sie kein Foto mittels eines Digitizers und einer Videokamera in Bewegung versetzen. Aber es gibt andere Möglichkeiten, die hier genutzt werden können. Ein Beispiel: Ihre Aufgabe ist es, einen Theatervorhang zu zeichnen, der nach den Seiten gerafft wird. Sie sollen nun eine Animationssequenz erstellen, die das Raffen der Vorhänge darstellt. Nun könnte man darangehen, per Maus einen Vorhang zu zeichnen und die einzelnen Phasen Stück für Stück zu erstellen. Ein Arbeitsaufwand, der einen versierten Computergrafiker gut eine Woche lang beschäftigen kann. Allerdings hat dieser Grafiker auch Videokamera und Digitizer benutzen können. Denn, nimmt man ein Stück Stoff und arrangiert es vor der Kamera so, daß in etwa der Anschein eines Theatervorhangs erweckt wird, können auf diese Art und Weise die einzelnen Grundphasen innerhalb von wenigen Stunden als Computergrafik vorliegen, die natürlich noch manuell bearbeitet werden müßten. Aber selbst wenn man diesen Zeitaufwand berücksichtigt, ist die Erstellung der Animation unter Zuhilfenahme der Videokamera weit effektiver. Solche Bilder können mit Comic Setter realisiert werden.

Gibt es neben der vernünftigen Software und der entsprechend leistungsfähigen Hard-

ware noch weitere Dinge, mit welchen sich unsere Amiga-Grafik-Workstation sinnvoll ergänzen ließe?

Immerhin gibt es da noch Lichtstifte, Trackballs und Grafiktablets. Lichtstifte haben sich in der Praxis bisher nicht bewähren können und Trackballs können, was die Präzision beim Positionieren des Cursors anbetrifft, nicht mit einer normalen Maus mithalten.

## Extrahardware für Deluxe Paint?

Grafiktablets sind ebenfalls recht ungenau und obendrein auch noch fast immer inkompatibel zu Deluxe Paint.

Tatsächlich hat sich in der Praxis die Maus ausgezeichnet bewährt. Wer an dieser Stelle noch Investitionen tätigen möchte, sollte sich also besser eine funktionelle Maus zulegen.

Das zukünftige Entwicklungen uns noch Eingabemedien beschere werden, die das Arbeiten mit Deluxe Paint noch komfortabler gestalten, ist nicht ausgeschlossen. Doch bis zur Stunde kann, außer der Maus, kein anderes wirklich überzeugen.

## Fotos vom Bildschirm

Kein Drucker kann ein computergrafisch erstelltes Bild in einer Qualität wiedergeben, die mit einem guten Bildschirmfoto konkurrieren könnte. Doch allein die Bildschirmfotografie ist alles andere als leicht. Wer von dieser Methode Gebrauch machen möchte, muß zunächst einmal über eine gewisse Ausrüstung verfügen. Dazu gehören neben einer Spiegelreflexkamera ein Stativ, und wenn möglich ein Fernauslöser.

Die besten Ergebnisse werden erzielt, wenn die Kamera zirka 1,5 Meter vom Monitor aufgestellt wird. Achten Sie darauf, daß die Fläche des Bildschirms im Verhältnis zum Kameraobjektiv möglichst parallel ausgerichtet ist, andernfalls kommt es auf dem Foto zu unschönen Verzeichnungen und Deformationen des Motives. Während der Aufnahme sollten möglichst alle anderen Lichtquellen im Raum abgeschaltet werden. Sollten Sie keine Rollos oder vergleichbares haben, ist es im Interesse eines lohnenden

Nürnberger Public Domain Studio  
Humboldtstr. 141 in CCC  
8500 Nürnberg 40  
Tel.: 0911 45 77 54 Fax: 0911 446 72 79

Ca. 9000

Public Domain + Shareware  
Disketten für

Amiga, Atari ST und PC.  
Gratis Katalog - Disk und Info  
für Ihren Computer anfordern!

## Farbige Disketten

	5.25" 2DD	
	10 Stück	100 Stück
Schwarz	DM5.90	DM54.90
Weiss	DM9.90	DM94.90
Rot	DM9.90	DM94.90
Grau	DM9.90	DM94.90
Gelb	DM9.90	DM94.90
Blau	DM9.90	DM94.90
Orange	DM9.90	DM94.90

	3.5" 2DD	
	10 Stück	50 Stück
Blau	DM11.90	-----
Grau	DM11.90	-----
Rot	DM17.90	DM85.00
Gelb	DM17.90	DM85.90
Grün	DM17.90	DM85.90

5.25" HD  
10 Stück 5.25" HD DM 12.80  
LEERDISKETTEN SIND NUR GEGEN VORZAHL  
ZU ERHALTEN

Versandkosten:

Nachnahme DM 6.- Vorkasse DM5.-

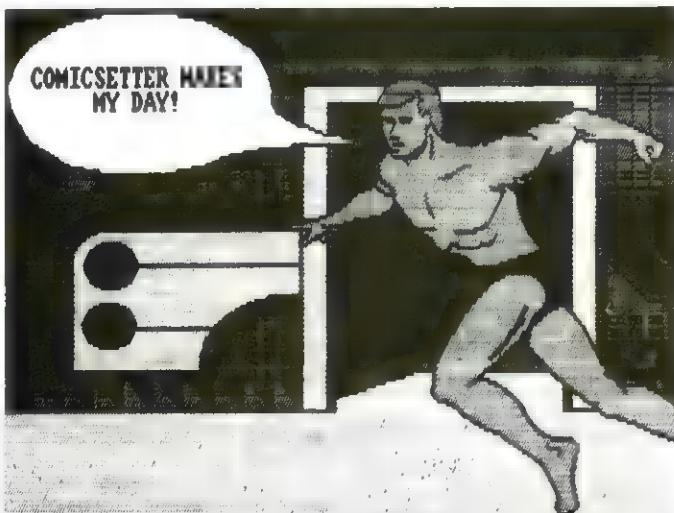
## Eine Bitte an unsere Abonnenten

Vermerken Sie bei  
Schriftverkehr und  
Zahlungen neben der  
vollständigen  
Anschrift stets Ihre  
Abo-Nummer.

Sie vermeiden damit  
unnötige Verzögerungen  
bei der  
Bearbeitung Ihres  
Abonnements.

Vielen Dank

Ihre DMV-  
Versandabteilung



Der Comic Setter, ein Grafikprogramm dessen Leistungen sich von Deluxe Paint unterscheiden

Bilderergebnisses sogar ratsam zu warten, bis es draußen dunkel geworden ist. Eine Eigenart der Bildröhre ist der große Anteil blauen Lichtes, den der Kathodenstrahl der Röhre erzeugt. So ist auch, wenn die Bildröhre nur einen weißen Bildschirm zeigt, immer ein sehr starker Blauanteil vorhanden. Ein Kunstlichtfilm mit entsprechenden Voraussetzungen kann Abhilfe schaffen. Zusätzlich ist auch der Helligkeitswert des Bildes ein Faktor, der berücksichtigt werden muß. In der Praxis hat sich herausgestellt, daß es ein Fehler ist, knauserig zu sein. Statt einem Foto pro Motiv sollten Sie mehrere mit verschiedenen Belichtungszeiten anfertigen. Zum einen ist dann sicher ein Bild gut geworden, zum anderen fallen viele Vergleichswerte an, anhand derer spätere Fotosessions leichter von der Hand gehen, und wahrscheinlich noch bessere Resultate zeitigen.

## Jenseits von Deluxe Paint

So leistungsstark und komfortabel Deluxe Paint ist, viele Grafikmöglichkeiten des Amigas bleiben diesem Programm verschlossen.

Wer mit dem Hold and Modify Mode oder dem Interlace Mode arbeiten will, muß auf andere Software zurückgreifen.

Während Deluxe Paint ein Bitmap-orientiertes Grafikprogramm ist, also in bestimmten Grenzfällen für die Darstellung eines Pixels fünf Bitmaps benötigt werden, gibt es auch objektorientierte Grafikprogramme. Hier wird jeder Zeichenvorgang nur auf einer Bitmap dargestellt und ist dementsprechend einfarbig. Da hier auch die Bildschirmauflösung weit höher gewählt werden kann (das Flimmern bleibt allerdings auch bei diesen Programmen und kann nur durch einen Fickerfixer unterdrückt werden), eignen sich diese Programme sehr gut zum Erstellen von Strichgrafiken, die dann im Rahmen von Desktop Publishing weitere Verwendung finden. Ein empfehlenswertes Programm dieser Art ist Professional Draw von Gold Disk.

Ein anderes recht eigenwilliges 'Grafikprogramm' ist der Comic Setter. Dieses Pro-

gramm ist ausschließlich auf das Erstellen von Computercomics ausgelegt. Neben der rein grafischen Seite bietet Deluxe Paint auch einige Animationsfeatures. Natürlich gibt es auch eine ganze Reihe von Programmen, die ausschließlich der Animation dienen. Eine sinnvolle Ergänzung von Deluxe Paint ist Zoetrope von GFA Systemtechnik.

## Animationen mit Deluxe Paint II?

Leider beinhaltet Deluxe Paint II nicht die guten Animationsfeatures der IIIer Version. Aber es gibt eine Möglichkeit, die auch den Deluxe-Paint-II-Anwendern Animationen ermöglichen. In der Ausgabe 4/90 der AMIGA DOS finden Sie ein Assemblerlisting, das einen Animator als zweiten Task neben Deluxe Paint eröffnet. Sie können diesen Animator während der Arbeit mit Deluxe Paint aufrufen und auf dem Deluxe-Paint-Zeichenbildschirm enthaltene Animationen ablaufen lassen.

## Finale

Unsere Deluxe-Paint-Werkstatt wurde zu einem ausgedehnten Streifzug durch die Wunderwelt der Amiga-Grafik. Sie haben gesehen, wie mit diesem Programm die tollsten Effekte gezaubert und die fantastischsten Bilder erstellt werden können. Jetzt sind Sie gefordert; Ihre Kreativität und Ihre Phantasie.

Auch wenn die Deluxe-Paint-Werkstatt mit diesem Teil schließt, werden wir uns diesem weiten Gebiet der Computeranimation in anderem Rahmen wieder nähern. Bis dahin wünschen wir Ihnen viel Spaß mit Deluxe Paint und Ihren Bildideen.

Im Rahmen unseres Pixel-Panoramas, das Sie auch in dieser AMIGA DOS finden, haben wir einen Grafikwettbewerb gestartet, an dem Sie sich mit Ihren Bildern beteiligen können. Wir freuen uns schon auf Ihren Beitrag.

(hs)



## Übersicht über die wichtigsten Deluxe Paint Tastaturkommandos

### Das Basismenü

b	Zeichenstifte definieren
B	---
c	Ungefüllte Kreise
C	Gefüllte Kreise
d	Kontinuierliches Freihandzeichnen
D	Gefülltes Freihandzeichnen
e	Ungefüllte Ellipse
E	Gefüllte Ellipse
f	Füllen
F	Füllart Requester
g	Gitter an/aus
G	Gitter an/aus, wobei die aktuelle Pinselposition als Schnittpunkt bestimmt wird
K	Bildschirm löschen
m	Vergrößerung an/aus
p	Farbpalette
q	Kurvenlinien
r	Ungefülltes Vieleck
R	Gefülltes Vieleck
s	Punktirtes Freihandzeichnen
t	Texteingabe
u	Letzte Änderung zurücknehmen
v	Gerade Linie
>	Vergrößerung - Maßstab größer
<	Vergrößerung - Maßstab kleiner
,	Farbwahl (Pick)

### Die Zeichenmodi

F1	Muster
F2	Farbe
F3	Stempel
F4	Schmierer
F5	Schatten
F6	Mischen
F7	Rollen
F8	Weich

### Animation und Tastatur

1	Ein Frame zurück
2	Ein Frame vorwärts
3	Gehe zu Frame Nr.
4	Kontinuierliches Abspielen
5	Einmaliges Abspielen
6	Animation vorwärts und rückwärts (Ping Pong) abspielen

### Brushbefehle auf der Tastatur

Z	Dehnen
h	Größe halbieren
H	Größe verdoppeln
x	Horizontal klappen
y	Vertikal klappen
z	90 Grad drehen
X	In horizontaler Richtung verdoppeln
Y	In vertikaler Richtung verdoppeln





Sebastian Ritter

# Quadrant 500

Alles auf einen Blick – Sieh zu, aber mach hin!

Bei unserem AmigaBASIC-Listing Quadrant 500 handelt es sich um ein Spiel, das eine schnelle Auffassungsgabe verlangt. Es gilt, innerhalb einer sich von Level zu Level dezimierenden Zeitspanne, eine bestimmte Farbkombination zu finden.

Nach dem Starten des Programms erscheint das Titelbild. Durch Betätigen der Feuertaste am Joystick gelangt man in die Highscore-Liste, die als separate Datei auf Diskette abgespeichert wird. Das Abspeichern dieser Liste erfolgt automatisch durch das Programm.

Wundern Sie sich also nicht, wenn irgendwann einmal eine Datei namens Hiscore auf Ihrer Diskette abgelegt wird. – Das ist alles ganz normal.

## Aufmerksamkeit ist alles

Mit der Tastenfunktion <Alt

S> läßt sich diese Highscore-Liste übrigens abspeichern, während durch gleichzeitiges Drücken der Tasten <Alt L> eine bereits abgespeicherte Liste eingeladen wird. Nun muß die Anzahl der Farben im Spiel eingegeben werden (Anm. der Red.: je mehr, desto leichter).

Zu Beginn des Spieles sieht man am rechten Bildschirmrand ein in zwölf mal zwölf Einheiten unterteiltes Quadrat. Jedes Kästchen dieser Fläche besitzt eine bestimmte Farbe. Am linken Bildrand ist ein drei mal drei Einheiten großes Quadrat, das einen Ausschnitt aus dem großen Quadrat anzeigt. Aufgabe des Spielers ist es nun, diese Farbsequenz des Ausschnitts mittels Cursor, der mit dem Joystick gesteuert wird, herauszusuchen und durch Drücken der Feuertaste zu fixieren. Das wäre ja recht einfach, wenn nicht...

## Wer sucht, der findet

...ja, wenn der Spieler unbegrenzte Zeit für diese Aktion hätte. Dem ist jedoch nicht so, denn nach 26 Sekunden (im ersten Level) ist "Ende im Gelände".

Wurde ein falscher Ausschnitt gewählt oder das Zeitlimit überschritten, so ist das Spiel beendet. Hat der Spieler währenddessen den korrekten Ausschnitt gefunden, wird er mit einem Bonus in Höhe der verbliebenen Zeit, multipliziert mit dem Faktor zehn, belohnt. Bei entsprechend hoher Punktzahl ist ein Highscore-Eintrag (maximal 16 Zeichen) möglich.

(br)

### Wichtig!!!

Die Zeilennummern am Anfang der Programmzeilen dienen nur der besseren Übersicht und sollten im BASIC-Editor nicht mit eingegeben werden. Sie können zur Eingabe auch den Checksummer CheckBAS aus Ausgabe 1/90 benutzen.

## Listings

```
000:998 REM *****
001:154 REM * QUADRANT 500 *
002:182 REM * Spiel *
003:916 REM * (c) 1990 by S.Ritter & AMIGA DOS *
004:612 REM * Sprache: AmigaBasic *
005:998 REM *****
006:000
007:577 DIM ta(15),feld(12,12),kfel(3,3),pun%(14),nam$(14),
      hi$(16)
008:706 FOR i%=1 TO 14
009:120 pun%(i%)=16000-i%*1000
010:018 nam$(i%)="Sebastian Ritter"
011:823 NEXT i%
012:288 RANDOMIZE TIMER
013:053 SCREEN 2,320,200,5,1
014:506 WINDOW 2,"Quadrant 500 (c) S. Ritter & AMIGA DOS",(-
      0,0)-(311,178),0,2
015:800 intro:
016:691 FOR i%=0 TO 31
017:064 PALETTE i%,0,0,0
018:823 NEXT i%
019:765 CALL ttitle(ta{}),a$)
020:182 sc%=0
Listing: Quadrant 500
```

```
021:543 lev%=1
022:627 CALL hiscoretabelle(pun%(),nam$(),sc%,hi$())
023:304 CALL spfarben
024:399 CALL farbenzahl(fa%)
025:936 spielfeldaufbau:
026:928 CALL spielfeld
027:704 FOR i%=1 TO 12
028:440 FOR i1%=1 TO 12
029:715 zf%=RND*fa%+1
030:601 feld(i%,i1%)=zf%
031:506 LINE((i%-1)*14+10,((i1%-1)*14+10)-((i1%-1)*14+16,
      ((i1%-1)*14+16),zf%+14,BF
032:707 NEXT i1%
033:823 NEXT i%
034:179 z1%=INT(RND*10)+1
035:947 z2%=INT(RND*10)+1
036:329 FOR i1%=1 TO 3
037:697 FOR i1%=1 TO 3
038:247 zf%=feld(z1%+i1%-1,z2%+i1%-1)
039:249 kfel(i%,i1%)=zf%
040:034 LINE((i%-1)*14+268,((i1%-1)*14+10)-((i1%-1)*14+27
      4,((i1%-1)*14+16),zf%+14,BF
041:707 NEXT i1%
042:823 NEXT i%
043:732 COLOR 28,4
044:855 LOCATE 14,23
Listing: Quadrant 500
```

# Listing

```

045:839 PRINT USING " Punkte:\ " ;STR$(sc%)
046:847 LOCATE 21,23
047:603 PRINT USING " Level :\ " ;STR$(lev%)
048:431 LOCATE 7,23
049:633 PRINT " Zeit:";SPACES$(4)
050:028 x%=6
051:036 y%=6
052:405 zeif%=27-lev%
053:342 IF zeif%<8 THEN zeif%=8
054:957 ON TIMER(1)GOSUB zeitkontrolle
055:898 LINE (x%,y%)-(x%+42,y%+42),28,b
056:158 WHILE INKEY$<>" "
057:196 WEND
058:100 hauptschleife:
059:608 FOR i=1 TO 50:NEXT
060:060 TIMER ON
061:437 IF zeif%<=0 GOTO levelende
062:543 fsv=STICK(2):mz=STICK(3):os=STRIG(3)
063:709 IF fsv=0 AND mz=0 AND os=0 GOTO hauptschleife
064:173 IF os=-1 GOTO levelende
065:894 h%=0
066:006 v%=0
067:978 IF mz=-1 AND y%>6 THEN v%=-14
068:458 IF mz=1 AND y%<132 THEN v%=14
069:402 IF fsv=1 AND x%<132 THEN h%=14
070:890 IF fsv=-1 AND x%>6 THEN h%=-14
071:773 IF h%=0 AND v%=0 GOTO hauptschleife
072:902 LINE (x%,y%)-(x%+42,y%+42),29,b
073:813 y%=y%+v%
074:641 x%=x%+h%
075:898 LINE (x%,y%)-(x%+42,y%+42),28,b
076:173 GOTO hauptschleife
077:000
078:125 SUB farben (ta()) STATIC
079:903 i%=1
080:185 FOR t=.9 TO .2 STEP-.1
081:026 ta(i%)=t
082:820 PALETTE i%,t,t,t
083:699 i%=i%+1
084:404 NEXT t
085:829 FOR t=.1 TO .9 STEP .1
086:026 ta(i%)=t
087:820 PALETTE i%,t,t,t
088:699 i%=i%+1
089:404 NEXT t
090:969 PALETTE 30,1,0,1
091:605 PALETTE 31,.33,.87,1
092:568 END SUB
093:000
094:421 SUB spfarben STATIC
095:706 FOR i%=1 TO 14
096:888 PALETTE i%,.25+(16-i%)*.05,0,0
097:823 NEXT i%
098:683 RESTORE farbdatas1
099:122 FOR i%=15 TO 30
100:971 READ a,b,c
101:427 PALETTE i%,a,b,c
102:160 NEXT
103:295 PALETTE 31,0,.93,.87
104:568 END SUB
105:000
106:373 SUB titel (ta(),a$) STATIC
107:503 CLS
108:878 f%=0
109:458 WHILE f%<178
110:705 FOR i%=1 TO 13
111:831 LINE (0,f%)-(312,f%),i%
112:711 LINE (0,f%+1)-(312,f%+1),i%
113:484 f%=f%+2
114:823 NEXT i%
115:196 WEND
116:430 LOCATE 7,14
117:470 COLOR 31,30
118:770 PRINT "Quadrant 500"
119:417 LOCATE 17,5
120:044 PRINT "1990 by S. Ritter ■ AMIGA DOS"
121:355 CALL farben (ta())
122:786 WHILE sgr=0
123:705 FOR i%=1 TO 13
124:441 FOR i1%=1 TO 13
125:808 IF STRIG(3)=-1 THEN EXIT SUB
126:865 t=ta((i%+i1%)MOD 13 +2)
127:932 PALETTE i1%,t,t,t
128:707 NEXT i1%
129:823 NEXT i%
130:196 WEND
131:568 END SUB
132:000
133:253 SUB spielfeld STATIC
134:878 f%=0
135:458 WHILE f%<178
136:706 FOR i%=1 TO 14
137:831 LINE (0,f%)-(312,f%),i%
138:483 f%=f%+1
139:823 NEXT i%
140:227 FOR i%=13 TO 1 STEP-1
141:831 LINE (0,f%)-(312,f%),i%
142:483 f%=f%+1
143:823 NEXT i%
144:196 WEND
145:342 FOR i%=6 TO 174 STEP 14
146:909 LINE (6,i%)-(173,i%),29
147:613 LINE (i%,6)-(i%,174),29
148:823 NEXT i%
149:966 FOR i%=6 TO 48 STEP 14
150:181 LINE(264,i%)-(306,i%),29
151:685 LINE(i%+258,6)-(i%+258,48),29

```

Listing: Quadrant 500

```

152:823 NEXT i%
153:568 END SUB
154:000
155:805 SUB hiscoretabelle (pun%(),nam$(),sc%,hi$()) STATIC
156:707 FOR i%=1 TO 15
157:272 PALETTE i%+14,.7,.09+i%/100*5,0
158:823 NEXT i%
159:838 a%=0
160:298 WHILE a%<178
161:027 FOR i%=29 TO 15 STEP-1
162:751 LINE (0,a%)-(312,a%),i%
163:351 LINE (0,a%+1)-(312,a%+1),i%
164:124 a%=a%+2
165:823 NEXT i%
166:144 FOR i%=16 TO 28
167:751 LINE (0,a%)-(312,a%),i%
168:351 LINE (0,a%+1)-(312,a%+1),i%
169:124 a%=a%+2
170:823 NEXT i%
171:196 WEND
172:000
173:284 namensausgabe:
174:136 PALETTE 31,.5,0,0
175:470 COLOR 15,22
176:391 LOCATE 2,15
177:612 PRINT "Highscores"
178:128 FOR i%=15 TO 28
179:503 COLOR 31,i%
180:615 LOCATE i%-10,7
181:271 PRINT SPACES$(27)
182:616 LOCATE i%-10,8
183:125 PRINT nam$(i%-14)
184:273 LOCATE i%-10,25
185:973 PRINT USING"#####";pun%(i%-14)
186:823 NEXT i%
187:637 IF sc%<pun%(14)GOTO tastendruck
188:477 COLOR 31,29
189:401 LOCATE 22,1
190:617 PRINT SPACES$(39);
191:401 LOCATE 22,1
192:487 PRINT "Bitte Namen eingeben:";
193:708 FOR i%=1 TO 16
194:603 hi$(i%)=INPUT$(1)
195:592 IF hi$(i%)=CHR$(13)GOTO einordnen
196:368 IF hi$(i%)=CHR$(31)AND i%>0 THEN i%=i%-2
197:854 IF i%<0 THEN i%=0
198:471 IF hi$(i%)<CHR$(32) OR hi$(i%)>CHR$(122) THEN i%=-1
199:854 IF i%<0 THEN i%=0
200:187 LOCATE 22,22+i%
201:361 PRINT hi$(i%);
202:823 NEXT i%
203:000
204:882 einordnen:
205:106 nam$(14)=""
206:708 FOR i%=1 TO 16
207:607 nam$(14)=nam$(14)+hi$(i%)
208:823 NEXT i%
209:855 pun%(14)=sc%
210:705 FOR i%=1 TO 13
211:003 p%=i%
212:170 FOR i1%=i%+1 TO 14
213:059 IF pun%(p%)<pun%(i1%)THEN p%=i1%
214:707 NEXT i1%
215:783 pun%(0)=pun%(i%)
216:579 pun%(i%)=pun%(p%)
217:377 pun%(p%)=pun%(0)
218:695 nam$(0)=nam$(i%)
219:987 nam$(i%)=nam%(p%)
220:081 nam%(p%)=nam$(0)
221:823 NEXT i%
222:182 sc%=0
223:401 LOCATE 22,1
224:617 PRINT SPACES$(39);
225:765 GOTO namensausgabe
226:000
227:060 tastendruck:
228:180 WHILE INKEY$<>" "WEND
229:674 a$=""
230:110 WHILE a$="" AND STRIG(3)=0
231:826 a$=INKEY$
232:196 WEND
233:926 i%=0
234:669 IF a$=CHR$(223)THEN GOSUB hiscoresave
235:234 IF a$=CHR$(163)THEN GOSUB hiscoreload
236:653 IF i%=1 THEN GOTO namensausgabe
237:503 CLS
238:640 EXIT SUB
239:812 hiscoresave:
240:027 OPEN"hiscore"FOR OUTPUT AS #1
241:706 FOR i%=1 TO 14
242:151 PRINT #1,pun%(i%)
243:567 PRINT #1,nam$(i%)
244:823 NEXT i%
245:043 CLOSE#1
246:982 RETURN
247:000
248:726 hiscoreload:
249:927 i%=1
250:283 OPEN"hiscore"FOR INPUT AS #1
251:706 FOR i%=1 TO 14
252:255 INPUT#1,pun%(i%)
253:671 INPUT#1,nam$(i%)
254:823 NEXT i%
255:043 CLOSE#1
256:982 RETURN

```

Listing: Quadrant 500



```

257:000
258:568 END SUB
259:000
260:293 SUB farbenzahl (fa%) STATIC
261:282   lin=0
262:668   FOR i%=0 TO 312 STEP 20
263:922     LINE (i%,0)-(i%+19,180),i%/20+14,BF
264:823   NEXT i%
265:000
266:030   fragebox:
267:930     LINE (0,80)-(312,100),1,BF
268:349   COLOR 9,1
269:396   LOCATE 11,8
270:316   PRINT "Wieviele Farben (2-16)?"
271:843   LOCATE 12,19
272:855   INPUT "",fa%
273:398   IF fa%<2 OR fa%>16 GOTO fragebox
274:487   fa%=fa%-1
275:668 END SUB
276:000
277:628   levelende:
278:174   TIMER OFF
279:530   k%=(x%+8)/14
280:706   l%=(y%+8)/14
281:076   FOR i%=k% TO k%+2
282:100     FOR j%=l% TO l%+2
283:552       IF feld(i%,j%)<>kfel(i%-k%+1,j%-l%+1)GOTO ver
284:707     NEXT j%
285:823   NEXT i%
286:253   sc=sc%+zei%*10+lev%
287:357   GOSUB punkteausdruckroutine
288:315   lev%=lev%+1
289:836   WHILE STRIG(3)=0:WEND
290:047   GOTO spielfeldaufbau
291:000
292:938   verloren:
293:902   LINE (x%,y%)-(x%+42,y%+42),29,b
294:058   LINE (z1%*14-8,z2%*14-8)-(z1%*14+34,z2%*14+34),28,b
295:516   COLOR 29,20
296:826   LOCATE 10,26
297:253   PRINT SPACES(10)
298:523   COLOR 29,19
299:834   LOCATE 11,26
300:442   PRINT " Verloren "
301:518   COLOR 29,22

```

Listing: Quadrant 500

```

302:842   LOCATE 12,26
303:253   PRINT SPACES(10)
304:408   FOR i%=600 TO 200 STEP-20
305:689     FOR j1%=0 TO 3
306:867       SOUND i%+j1%*15,2,255,0+j1%
307:707     NEXT j1%
308:823   NEXT i%
309:836   WHILE STRIG(3)=0:WEND
310:503   CLS
311:627   CALL hiscoretabelle (pun%(),nam%(),sc%,hi%())
312:563   GOTO intro
313:000
314:628   punkteausdruckroutine:
315:856   LOCATE 14,32
316:047   PRINT USING"\          \";STR$(sc%)
317:848   LOCATE 21,32
318:227   PRINT USING"\          \";STR$(lev%)
319:702   COLOR 13,2
320:193   LOCATE 7,5
321:115   PRINT SPACES(9)
322:706   COLOR 13,6
323:197   LOCATE 8,5
324:966   PRINT "  Richtig  "
325:690   COLOR 2,10
326:201   LOCATE 9,5
327:115   PRINT SPACES(9)
328:771   FOR i%=255 TO 0 STEP-15
329:388     SOUND 262,2,i%,0
330:877     SOUND 330,2,i%,1
331:670     SOUND 392,2,i%,2
332:031     SOUND 523,2,i%,3
333:823   NEXT i%
334:982   RETURN
335:000
336:404   zeitkontrolle:
337:671   ze1%=ze1%-1
338:431   LOCATE 7,31
339:363   PRINT USING"##";ze1%
340:982   RETURN
341:000
342:184   farbdatas1:
343:943   DATA .4,.6,1,0,0,1,1,.6,.67,.3,.3,.3
344:551   DATA 0,1,0,.4,.7,.5,1,.82,0,.1,.5,.15
345:173   DATA 1,0,1,1,.6,1,.3,.4,.5,.8,.6,.53
346:545   DATA .5,.3,.7,1,1,1,0,0,0,.5,.5,.5

```

Listing: Quadrant 500

## TOOLBOX EDITION 68000

Wer seinen Computer oder Anwendungen programmieren will, findet hier Tips, Anregungen und vor allem Anwendungen erster Güte.

Das Projekt Fraktal wartet mit einem Weltrekord auf: Das "Apfelmännchen" wird in 7 Sekunden auf den Bildschirm eines Amiga 2000 gezeichnet!! Da waren Hexenmeister am Werk.

Viele Reviews informieren über neue Programmierer-Software und helfen dem Leser zum Überblick über interessante Neuerungen.

Natürlich gibt es die Programme des Heftes auch in der DATABOX. DATABOX ist der Software-Service, der bei DMV Standard ist. Wem das Tippen zu fehlerträchtig ist, der bestellt die DATABOX zum Heft.

### Aus dem Inhalt:

PROJEKT FRAKTAL – Portierung in M2Amiga  
 WELTREKORD – Apfelmännchen in 7 Sekunden  
 ACU – Mausgesteuerter Compiler  
 C-Compiler-Oberfläche  
 Universeller Multitasking-FileRequester in M2Amiga  
 Undercover Viruskiller mit eigenem Task  
 Do-Request – Trickreicher AutoRequester  
 AmigaDOS – Fenster mit Zwittereigenschaften  
 M2Amiga-Inline-Code per Programm

**DATABOX Amiga** 29,- DM\*

### TOOLBOX EDITION 68000

**Sonderpreis für AMIGA-DOS-Leser** 9,- DM\*

**Paketpreis Edition 68000 + DATABOX Amiga** 29,- DM\*

\* Unabhängig von der Anzahl der bestellten Produkte berechnen wir für das Inland 4,- DM bzw. für das Ausland 6,- DM Porto und Verpackung.

**DMV-Verlag • Postfach 250 • 3440 Eschwege**

Nr. 1/88-89

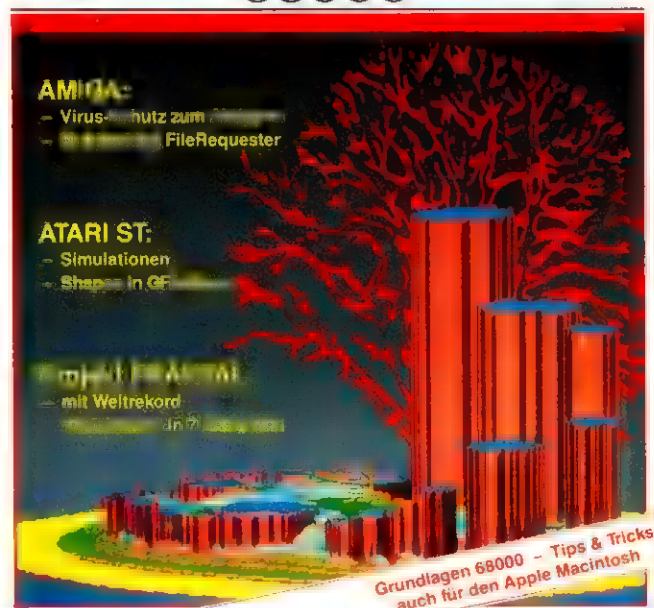
DK - DM 145 ÖS 18 sfr

DMV Verlag

## EDITION

# TOOLBOX

• Amiga • Atari • 68000 • Macintosh •



Monika Schmid

## Meister Klecksel unterwegs

### Farben ins CLI gebracht

Vorbei sind die Zeiten, wo der Kommandozeilen-Interpreter als kümmerliches graues Mäuschen sein Dasein ohne jegliche Farbtupfer fristete. Ein gutes AmigaDOS-Handbuch, ein wenig Überlegung, und schon kann's losgehen. Tricksen wir ein wenig mit dem CLI herum.

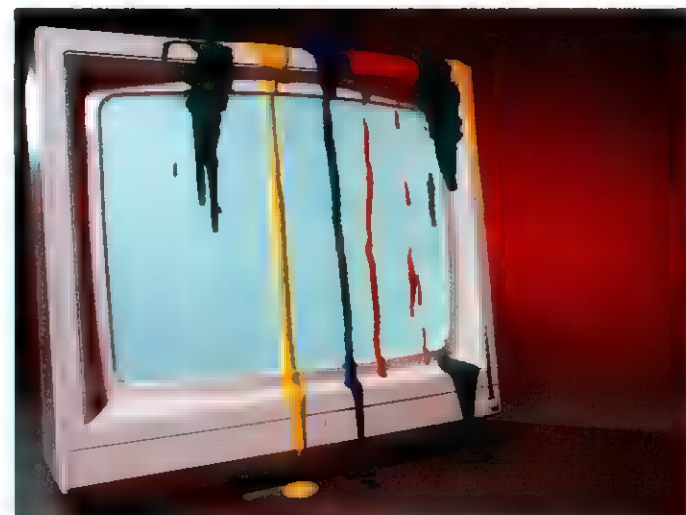
In Kürze noch einiges zum CLI: Möchte man sich nicht auf die Ebene der Workbench begeben, sondern einen noch etwas innigeren Kontakt zum Rechner herstellen, begibt man sich auf die Ebene des CLI. Direkt nach dem Start des Amiga kommt man zum ersten Mal mit dem CLI, CommandLineInterpreter, in Berührung. Das erste CLI-Fenster gibt die Meldungen des DOS aus. CLI stellt also die direkte Beziehung zwischen dem Anwender, der seine Befehle eingibt, dem DOS, das diese Befehle maschinengerecht aufbereitet und dem Rechner selbst, der diese Befehle verarbeitet, her.

Das CLI selbst ist an und für sich recht farblos – im wahren Sinne des Wortes. Eigentlich ist es die einfachste Sache

der Welt, Farben in den CLI zu bringen. Ein kleiner Trick hilft da weiter. Man muß nur die Steuersequenzen aus dem AmigaDOS-Handbuch eintippen, RETURN drücken und das "unknown command" über sich ergehen lassen. Schon kann man im CLI mit der gewünschten Vordergrund- oder Hintergrundfarbe weiter-schreiben. Ebenso lassen sich die Styles normal, fett, unterstrichen und kursiv darstellen, wobei kursiv erst dann richtig zur Geltung kommt, wenn der Transparent-Modus eingestellt ist.

Einige Sequenzen, wie man beispielsweise Styles verändert: Setzt man

- ESC[0m RETURN ein, kann man alle eingestellten Werte wieder zurücksetzen, oder mit
- ESC[3m RETURN



eine kursive Ausgabe erreichen oder

■ ESC[4m RETURN den Ausgabertext unterstrichen darstellen oder

● ESC[1m RETURN den Ausgabertext in fetter Schrift darstellen.

Mit ESC ist dabei nicht der Text ESC gemeint, sondern die Escape-Taste. Ebenso ist mit RETURN nicht der Text "RETURN" gemeint, sondern die RETURN-Taste. Diese Informationen können Sie dem AmigaDOS-Handbuch (ganz hinten bei den Drucker-Steuersequenzen) entnehmen. Für Amiga-Einsteiger ist es sicherlich nicht bekannt, daß man diese Steuersequenzen direkt ins CLI-Fenster eintippen kann, um den Farb-

oder Stilwechsel herbeizuführen. Die anderen Steuersequenzen, die weder Farbe noch Stil betreffen, funktionieren im CLI allerdings nicht, soweit ich das ausprobieren habe.

Aus einigen "Try and Error"-Erfahrungen entstand nun ein kleines Assembler-Programm, das nach der gewünschten Farbe fragt und diese dann einstellt.

Das Ganze wurde mit dem DevPac-Assembler erstellt.

(vb)

#### Literatur

Commodore Amiga Inc.,  
AmigaDOS 1.3, ISBN  
3-89090-802-0,  
Markt&Technik Verlag,  
1989, München

#### Listings

```
1: *****
2: * bunt.asm
3: * Dieses Programm bringt Farben in den CLI
4: * (c) 1990 M. Schmid & AMIGA DOS
5: * Sprache: Devpac Assembler
6: *****
7:
8: include "sys:exec/exec_lib.i" ; ggfs. anpassen
9: include "sys:libraries7/dos_lib.i" ; ggfs. anpassen
10:
11: movem.l d0-d7/a0-a6,-(sp) ; Register retten
12: lea dosname,a1 ; Lib.Name nach A1
13: move.l #0,d0 ; Version nach D0
14: call exec OpenLibrary
15: tst.l d0
16: beq Fehler
17: move.l d0,_Dosbase ; Speichern Dos-Zeiger
18:
19: call dos Output ; suche Filehandler
20: des aktuellen Fensters
21: tst.l d0
22: beq aus
23: move.l d0,Aktuellesfenster
24:
25: Fragen1:
26: move.l #0,Antwort1
27: move.l Aktuellesfenster,d1
28: move.l #Bittel,d2 ; Vordergrundfarbe ?
29: move.l #25,d3
30: call dos write
31:
32: Eingabe1:
33: move.l Aktuellesfenster,d1
34: move.l #1,d2 ; auf Eingabe warten
35: call dos WaitForChar
36: tst.l d0
37: beq Eingabe1
38:
39: Eingabe2:
40: move.l Aktuellesfenster,d1
41: clr.l d0
```

Listing: bunt.asm

```
41: move.l #Antwort1,d2
42: ;dahinein sollen die Zeichen geschrieben werden
43: move.l #70,d3
44: call dos read
45: ; 70 Zeichen aus Tastatur in den Datenpuffer schreiben
46: cmpi.l #1,d0 ; Ende bei Fehler
47: beq aus
48:
49: ; Anzahl Zeichen bis
50: cmpi.l #2,d0 ; zum RETURN gelesen
51: beq rechtl
52: bsr zulang ; oder zu kurz
53: bra Fragen1
54: rechtl:
55: move.l #Antwort1,a0
56: move.b (a0),d7 ; Ziffer holen
57: cmpi.b #0,d7
58: beq rechtl
59: cmpi.b #1,d7
60: beq rechtl
61: cmpi.b #2,d7
62: beq rechtl
63: cmpi.b #3,d7
64: beq rechtl
65: bsr wiebitte
66:
67: rechtl:
68: bra Fragen1
69:
70: Fragen2:
71: move.l #0,Antwort2
72: move.l Aktuellesfenster,d1
73: move.l #Bittel2,d2
74: move.l #25,d3
75: call dos write
76:
77: Eingabe3:
78: move.l Aktuellesfenster,d1
79: move.l #1,d2 ; auf Eingabe warten
80: call dos WaitForChar
81: tst.l d0
82: beq Eingabe3
83: Eingabe4:
```

Listing: bunt.asm



```

84:      move.l    Aktuellesfenster,d1
85:      clr.l     d0
86:      move.l    #Antwort2,d2 ; Zeichen werden da
87: ; hinein geschrieben
88:      move.l    #70,d3
89:      callDOS   read
90: ; 70 Zeichen von Tastatur in Datenpuffer schreiben
91:      cmpi.l    #-1,d0 ; Ende bei Fehler
92:      beq       aus
93: ; so viele Zeichen konnten bis zum
94: ; RETURN gelesen werden
95:
96:      cmpi.l    #2,d0 ; = 2 Zeichen
97:      beq       recht3
98:      bsr       zulang ; oder zu kurz
99:      bra       Fragen2
100: recht3:
101:      move.l    #Antwort2,a0
102:      move.b    (a0),d7 ; Ziffer holen
103:      cmpi.b    #'0',d7
104:      beq       recht4
105:      cmpi.b    #'1',d7
106:      beq       recht4
107:      cmpi.b    #'2',d7
108:      beq       recht4
109:      cmpi.b    #'3',d7
110:      beq       recht4
111:      bsr       wiebitte
112:      bra       Fragen2
113: recht4:
114: vergleichen:
115:      move.l    #Antwort1,a1
116:      move.b    (a1),d1
117:      move.l    #Antwort2,a2
118:      move.b    (a2),d2
119:      cmp.b     d1,d2
120:      beq       blind
121:
122: Setzen0:
123:      move.l    #Antwort1,a2
124:      move.b    (a2),d2
125:      cmpi.b    #'0',d2
126:      bne       setzen1
127:      move.l    Aktuellesfenster,d1
128:      move.l    #SetAPen0,d2
129:      move.l    #6,d3
130:      callDOS   write
131:      bra       Hinter0
132:
133: Setzen1:
134:      move.l    #Antwort1,a2
135:      move.b    (a2),d2
136:      cmpi.b    #'1',d2
137:      bne       Setzen2
138:      move.l    Aktuellesfenster,d1
139:      move.l    #SetAPen1,d2
140:      move.l    #6,d3
141:      callDOS   write
142:      bra       Hinter0
143:
144: Setzen2:
145:      move.l    #Antwort1,a2
146:      move.b    (a2),d2
147:      cmpi.b    #'2',d2
148:      bne       Setzen3
149:      move.l    Aktuellesfenster,d1
150:      move.l    #SetAPen2,d2
151:      move.l    #6,d3
152:      callDOS   write
153:      bra       Hinter0
154:
155: Setzen3:
156:      move.l    #Antwort1,a2
157:      move.b    (a2),d2
158:      cmpi.b    #'3',d2
159:      bne       Hinter0 ; eigentlich unmöglich
160:      move.l    Aktuellesfenster,d1
161:      move.l    #SetAPen3,d2
162:      move.l    #6,d3
163:      callDOS   write
164:
165: Hinter0:
166:      move.l    #Antwort2,a2
167:      move.b    (a2),d2
168:      cmpi.b    #'0',d2
169:      bne       Hinter1
170:      move.l    Aktuellesfenster,d1
171:      move.l    #SetBPen0,d2
172:      move.l    #8,d3
173:      callDOS   write
174:      bra       Ende
175:
176: Hinter1:
177:      move.l    #Antwort2,a2
178:      move.b    (a2),d2
179:      cmpi.b    #'1',d2
180:      bne       Hinter2
181:      move.l    Aktuellesfenster,d1
182:      move.l    #SetBPen1,d2
183:      move.l    #8,d3
184:      callDOS   write
185:      bra       Ende
186:
187: Hinter2:
188:      move.l    #Antwort2,a2
189:      move.b    (a2),d2
190:      cmpi.b    #'2',d2
191:      bne       Hinter3
192:      move.l    Aktuellesfenster,d1
193:      move.l    #SetBPen2,d2
194:      move.l    #8,d3
195:      callDOS   write
196:      bra       Ende
197:
198: Hinter3:
199:      move.l    #Antwort2,a2
200:      move.b    (a2),d2
201:      cmpi.b    #'3',d2
202:      bne       Ende ; darf eigentlich nicht sein
203:      move.l    Aktuellesfenster,d1
204:      move.l    #SetBPen3,d2
205:      move.l    #8,d3
206:      callDOS   write
207:
208: Ende:
209: ; ** hier evtl. Probetext zu Testzwecken ausgeben ***
210:

```

Listing: bunt.asm

```

204: aus:
205:      move.l    _Dosbase,a1
206:      callExec  CloseLibrary
207:
208: Fehler:
209:      movem.l    (sp)+,d0-d7/a0-a6 ; Register zurück
210:      rts       ; Rück-> Programmende
211:
212: ; ***** Hauptprogrammfragmente *****
213:
214: blind:
215:      move.l    #0,Antwort3
216:
217: Fragen3:
218:      move.l    Aktuellesfenster,d1
219:      move.l    #Unsichtbar,d2
220:      move.l    #46,d3
221:      callDOS   write
222:
223: Eingabe5:
224:      move.l    Aktuellesfenster,d1
225:      move.l    #1,d2 ; auf Eingabe warten
226:      callDOS   WaitForChar
227:      tst.l     d0
228:      beq       Eingabe5
229:
230: ; ***** Dateinamen eingegeben wurde *****
231:
232: Eingabe6:
233:      move.l    Aktuellesfenster,d1
234:      clr.l     d0
235:      move.l    #Antwort3,d2
236: ;dahinein sollen die Zeichen geschrieben werden
237:      move.l    #70,d3
238:      callDOS   read
239: ; 70 Zeichen von der Tastatur in Datenpuffer schreiben
240:      cmpi.l    #-1,d0 ; Ende bei Fehler
241:      beq       aus
242: ; so viele Ziffern konnten bis
243: ; zum RETURN gelesen werden
244:      cmpi.l    #2,d0 ; = 2 Zeichen
245:      beq       recht5
246:      bsr       zulang ;oder zu kurz
247:      bra       Fragen3
248:
249: recht5:
250:      move.l    #Antwort3,a0 ;Ziffer holen
251:      move.b    (a0),d7
252:      cmpi.b    #'3',d7
253:      beq       recht6
254:      cmpi.b    #'n',d7
255:      beq       recht6
256:      bsr       wiebitte
257:      bra       Fragen3
258:
259: recht6:
260:      cmpi.b    #'n',d7
261:      beq       Fragen1
262:      bra       Setzen0
263:
264: ; ***** Unterprogramme *****
265:
266: zulang:
267:      move.l    Aktuellesfenster,d1 ; Return
268:      move.l    #Lang,d2
269:      move.l    #1,d3
270:      callDOS   write
271:
272: wiebitte:
273:      move.l    Aktuellesfenster,d1
274:      move.l    #Bahnhof,d2
275:      move.l    #25,d3
276:      callDOS   write
277:      rts
278:
279: ; ***** Definitionen *****
280:
281: dosname:
282:      dc.b      "dos.library",0
283:      even
284:
285: _Dosbase:
286:      dc.l      0
287:
288: Antwort1:
289:      ds.b      80 ; max. 80 Bytes
290:
291: Antwort2:
292:      ds.b      80 ; max. 80 Bytes
293:
294: Antwort3:
295:      ds.b      80 ; max. 80 Bytes
296:
297: Bitte1:
298:      dc.b      "Vordergrundfarbe ? (0-3) ",0
299:      even
300:
301: Bitte2:
302:      dc.b      "Hintergrundfarbe ? (0-3) ",0
303:      even
304:
305: Aktuellesfenster:
306:      dc.l      0
307:      even
308:
309: Lang:
310:      dc.b      $0A,0
311:      even
312:
313: Bahnhof:
314:      dc.b      "Das verstehe ich nicht !",10,0
315:      even
316:
317: Unsichtbar:
318:      dc.b      "Wollen Sie wirklich unsichtbar we
319:      rden (j/n) ? ",0
320:      even
321:
322: SetAPen0:
323:      dc.b      27,"[30m",0 ; Penfarbe auf 0
324:
325: SetAPen2:
326:      dc.b      27,"[32m",0 ; Penfarbe 2
327:
328: SetAPen3:
329:      dc.b      27,"[33m",0 ; Penfarbe 1
330:
331: SetBPen1:
332:      dc.b      27,"[41m",10,0,0 ; Paper 1
333:
334: SetBPen2:
335:      dc.b      27,"[42m",10,0,0 ; Paper 2
336:
337: SetBPen3:
338:      dc.b      27,"[43m",10,0,0 ; Paper 3
339:
340: SetAPen1:
341:      dc.b      27,"[31m",0 ; Penfarbe 1
342:
343: SetBPen0:
344:      dc.b      27,"[40m",10,0,0 ; Paper 0
345:
346: ; ***** Ende von bunt.asm *****

```

Listing: bunt.asm

**Endlich keine Listings  
mehr abtippen!**

Nicht bei allen Programmen ist es mit drei Zeilen getan – gute Routinen und praktische Funktionen brauchen ihren Platz. Und bisweilen lassen sich auch lange Datenblöcke nicht vermeiden, ganz zu schweigen von Hexdumps und Assemblerlistings. Schonen Sie Ihre Augen und schlagen Sie sich nicht die Nacht mit Abtippen um die Ohren. – Auf der Databox zum Amiga DOS-Heft finden Sie alle Listings als ASCII-File: passend für jeden Texteditor, den Amiga-BASIC-Interpreter, Makro-Assembler oder einen Compiler für C und Modula-2.

**Alle Programme sofort  
nutzen**

Da ist er nun endlich – der Trick oder das Programm, auf das Sie schon so lange gewartet haben! Zu allem Unglück ist das Listing aber in Modula-2 oder C, jedenfalls in einer Sprache, zu der Sie keinen Compiler haben, um ein lauffähiges Programm herzustellen.

Auch in diesem Fall hilft Ihnen die Databox von Amiga DOS aus der Patsche: Neben den Quelltexten im ASCII-Format finden Sie jeweils auch das fertige, lauffähige Programm. Sie brauchen es nur von der Databox-Diskette aus zu starten.

# DATA

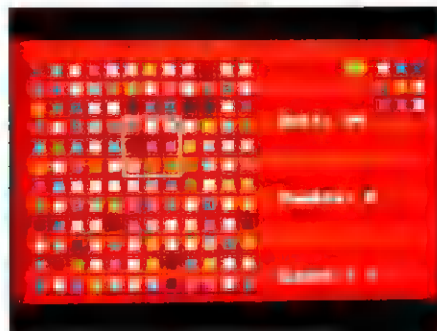
**Keinen Ärger mehr mit  
Tippfehlern**

Wer kennt das nicht, wenn das Programm nach dem Eintippen nicht läuft oder der Rechner gar abstürzt. Besonders gemein sind auch Fehler, die erst nach Wochen bei einer bisher nicht gebrauchten Funktion zu Tage treten, oder wenn der Druckfehlerteufel am Werke war.

Zermartern Sie sich nicht den Kopf, bis Sie die falsche Zahl im Datafeld gefunden haben. – Alle Dateien auf der Databox zur Amiga DOS sind vom Autor und der Redaktion auf Fehlerfreiheit geprüft und im dazugehörigen System "probegelaufen".



Zaubern Sie sich mit unserem Fraktal-Generator in eine faszinierende Grafikwelt



Um sich bei dem AmigaBASIC-Listing Quadrant 500 in die HighScore-Liste eintragen zu können, bedarf es fast Adleraugen



Alle Listings und Programme auf Diskette –  
Computer einschalten – Diskette einlegen –  
los geht's

2 Disketten  
voller  
Programme

# BOB X



Unser Filemonitor ist ein  
Utility zum Einlesen und  
Editieren von Program-  
men. Dieses Tool darf  
bei keinem Amiga-  
Programmierer fehlen

24,- DM

Wenn Sie über den DMV-Bestellservice bestellen, gilt folgendes:

Inland:		Ausland:	
Einzelpreis	24,- DM	Einzelpreis	24,- DM
zzgl. Versandkosten	4,- DM	zzgl. Versandkosten	6,- DM
<b>Endpreis</b>	<b>28,- DM</b>	<b>Endpreis</b>	<b>30,- DM</b>

#### Zahlungsweise:

Am einfachsten per Vorkasse (Verrechnungsscheck) oder als Nachnahme zuzü-  
glichen der Nachnahmegebühr. (Bei Lieferungen in das Ausland ist Nachnahme  
nicht möglich.)

Bitte benutzen Sie die Bestellkarte.

DMV-Verlag · Postfach 250 · 3440 Eschwege



Basic

C

Modula-2

Assembler

Lauffähiges Programm

#### Inhalt der Databox AMIGA DOS 5/90

- x BASIC  
Quadrant 500 – ein Spiel bei dem es  
auf eine schnelle Auffassungsgabe an-  
kommt
- x Der AmigaBASIC-Kurs – wie immer  
alle Beispielprogramme lauffähig auf  
der Databox
- x C  
Das Spriteeditor-Projekt – Alle vier  
Teile in einem Listing vereint. Das un-  
bedingte Muß zu professionellen Er-  
stellung von Sprites, BOBs und Images
- x Modula-2  
Einbinden von Maschinencode in  
M2Amiga
- x Assembler  
Beispielprogramme zur Blitter- und  
Copperprogrammierung
- x x Aufsteigerwissen Assembler:  
Paintbox – es geht weiter
- x x Der FileMonitor zum Ansehen und  
Editieren von Dateien
- x x Ein Generator zur Berechnung und Er-  
stellung wunderschöner Fraktale
- x x LinkChecker und LinkList – Die ulti-  
mative Vorbeugung gegen Linkviren



Der Faszination fraktaler Grafiken kann sich kaum jemand entziehen. Wir präsentieren Ihnen den schnellsten Fraktalgenerator für den Amiga, der je abgedruckt wurde.

Peter Woop

## Faszination Fraktal

**Wer Mandelbrotfiguren berechnen will, ohne darauf mehrere Stunden warten zu wollen, muß sich auch auf einem nicht gerade langsamen Rechner wie dem Amiga etwas einfallen lassen. Unser Listing zeigt, wie's schnell und benutzerfreundlich geht.**

**A**ls Programmiersprache wurde einmal mehr Assembler gewählt, da nur so einige der elementaren Ideen dieses Programms zu verwirklichen waren. Allerdings sieht man auch an der Länge des Listings, daß Assemblerprogramme (zumindest was den Quellcode betrifft) meist um einiges länger sind als in Hochsprachen entwickelte. Aber was tut man nicht alles, um etwas weniger lange warten zu müssen.

### Die Funktionen

Doch nun zur Funktionsweise: Das Programm berechnet die zum Setzen der Punkte benötigte Adresse nicht jedesmal neu, sondern füllt den Bildschirm von oben nach unten. Dabei wird einfach das entsprechende Adreßregister, das auf den Speicher zeigt, fortlaufend erhöht. Auch muß

kein unnötiger Aufwand damit betrieben werden, ein Wort aus dem Speicher zu holen und mit den zu setzenden Bits zu verknüpfen, um dann wieder zu schreiben. Statt dessen lassen sich die 16 Bits eines Wortes per Oder-Befehl direkt mit dem Speicher verknüpfen, da jeder Punkt nur einmal gesetzt wird und somit vor dem Setzen gleich null ist.

### Jetzt kommt Farbe ins Spiel

Auch bei der Farbwahl wurde optimiert: Hier wird für jede der 16 möglichen Farben in eine eigene Routine verzweigt, die nur die Bitplanes beschreibt, bei denen dies wirklich notwendig ist. Da bei diesem Verfahren die Bitplanes unbedingt hintereinander liegen müssen, wird nach dem Programmstart ein zusammenhängender Spei-

cherblock reserviert, der den eigentlichen, möglicherweise auseinandergerissenen. Speicher ersetzt. Eine weitere Geschwindigkeitssteigerung läßt sich durch eine bestimmte Eigenschaft der Mandelbrotmenge erreichen: Liegen nämlich zwei Punkte nur nahe genug beieinander, so liefern sie auch dasselbe Ergebnis. Das Programm testet zunächst, ob in einer Entfernung von acht Bildschirmpunkten immer noch derselbe Funktionswert geliefert wird. Ist dies der Fall, werden acht Punkte der entsprechenden Farbe gesetzt, ansonsten reduziert sich die Schrittweite auf die Hälfte. Im schlechtesten Fall werden alle Punkte wieder einzeln berechnet. Allerdings hat auch dieses Verfahren einen Haken, denn sollte sich der Funktionswert zwischen zwei Punkten mit demselben Wert kurz ändern, so bleibt dies unberücksich-

tigt. Obwohl dieses Phänomen im Verhältnis zum gesamten Bildschirm nur recht selten auftritt, trägt es nicht gerade zur Ansehnlichkeit der Bilder bei. Aus diesem Grund wurde eine weitere Funktion implementiert, mit der der Benutzer bestimmte Teile eines Bildes ohne diese Sprungtechnik, also auch ohne mögliche Fehler, nachträglich neu berechnen kann.

### Präzision mit 32 Bit

Durch die Verwendung einer weiteren Iterationsroutine mit geringerer Rechengenauigkeit als der im Programm verwendeten (32 Bit) ließe sich die Geschwindigkeit noch weiter steigern. Weil aber mit einer derartigen Routine nicht viel mehr als die Berechnung der Urfigur möglich ist, wurde diese Idee schnell wieder verworfen. Auch wenn sich die Programmierung kompliziert anhören mag, die Bedienung des Programms ist es keineswegs: Nach dem Start erscheint ein Fenster auf dem CLI-Bildschirm, das alle erforderlichen Bedienelemente enthält. Dort findet man unter anderem einige String-Gadgets, in die man nach der Aktivierung per Mausklick die zur Berechnung des Mandelbrotbildes nötigen Werte wie Koordinaten und Rechentiefe eingeben kann. Dabei ist das Intervall des Realteils (x-Richtung) und des Imaginärteils (y-Richtung) in derselben Form anzugeben, in der man die Anfangswerte (jeweils von -3,5 bis +3,5) nach dem Start vorfindet. Dies ist erforderlich, da eine komplexe Überprüfung auf Eingabefehler das Programm noch weiter in die Länge gezogen hätte. Allerdings braucht kein führendes Pluszeichen angegeben zu werden, ebenso wie Nullen am Ende der Zahl weggelassen können. Die Rechentiefe gibt die Anzahl der Iterationen an, nach denen spätestens abgebrochen werden soll. Mit dem Anklicken von "Bild berechnen" läßt sich schließlich das Fraktal errechnen, das übrigens auf einem zweiten Bildschirm angezeigt wird. Obwohl das Programm nach jeder Berechnung wieder zum CLI-Bildschirm umschaltet, läßt sich natürlich jederzeit mit der linken Amiga-Taste und M oder N zwischen den Screens hin- und herspringen.



Bei "Ausschnitt" erscheint ein Fadenkreuz, mit dem per Maus der Bereich des Bildes markiert werden kann, der später vergrößert berechnet werden soll.

## Im Fadenkreuz: fraktale Grafiken

Dieser Vorgang läßt sich übrigens beliebig oft wiederholen, da die Koordinaten erst

dann endgültig übernommen werden, wenn ein Bild berechnet wird. "Polieren" erlaubt in ähnlicher Weise die Auswahl eines Bildbereiches. Allerdings bleibt hier das berechnete Bild bestehen, und nur dieser Bereich wird, wie weiter oben schon erwähnt, nochmals Punkt für Punkt errechnet. Schließlich gestattet die Funktion "Bild speichern"

das Abspeichern eines errechneten Fraktals auf Diskette, wobei das IFF-Format Verwendung findet. Allerdings wurde (auch hier aus Platzgründen) auf den Pack-Algorithmus verzichtet, so daß das Bild unkomprimiert gespeichert wird. Hierbei ist zu berücksichtigen, daß jedes Bild etwa 41 kByte in Anspruch nimmt und natürlich entspre-

chend viel Platz auf der Diskette vorhanden sein muß. Ansonsten bleibt uns nur noch, Ihnen viel Spaß beim Errechnen der Bilder zu wünschen. Im übrigen sollte es kein allzu großes Problem darstellen, nicht implementierte Funktionen wie Packer, Farbeinsteller oder Überprüfung von Eingabefehlern "nachzurüsten". (mm)

## Listings

```
***** AMIGA DOS INFO *****
* (C)1990 by Peter Woop - AMIGA DOS
* Name : Fraktalgenerator
* Sprache : ASSEMBLER
* Besonderheiten : keine
*****

1:  ----- library-offsets -----
2:
3:  execbase = 4
4:
5:  allocmem = -198
6:  freemem = -210
7:  getmsg = -372
8:  closelibrary = -414
9:  openlibrary = -552
10:
11:  closescreen = -56
12:  closewindow = -72
13:  displaybeep = -96
14:  openscreen = -198
15:  openwindow = -204
16:  refreshgadgets = -222
17:  screentofront = -252
18:  wbenchtofront = -342
19:  remakedisplay = -384
20:
21:  clearscreen = -48
22:  move = -240
23:  draw = -246
24:  writepixel = -324
25:  setapen = -342
26:  setdrmd = -354
27:
28:  open = -30
29:  close = -36
30:  write = -48
31:
32:  ----- programmstart -----
33:
34:  beginning:
35:      move.l  execbase,a6
36:
37:  ----- benoetigte libraries oeffnen -----
38:
39:
40:  lea  dosname,a1      ; dos-library
41:  moveq #0,d0
42:  jsr  openlibrary(a6)
43:  move.l  d0,dosbase   ; zeiger sichern
44:  beq  no_dos          ; bei fehler programmende
45:
46:  lea  intname,a1      ; intuition-library
47:  moveq #0,d0
48:  jsr  openlibrary(a6)
49:  move.l  d0,intbase
50:  beq  no_int
51:
52:  lea  gfxbase,a1      ; graphics-library
53:  moveq #0,d0
54:  jsr  openlibrary(a6)
55:  move.l  d0,gfxbase
56:  beq  no_gfx
57:
58:  ----- screen und window oeffnen -----
59:
60:  move.l  intbase,a6
61:  lea  screen_defs,a0
62:  jsr  openscreen(a6)
63:  move.l  d0,screen_hd
64:  beq  no_screen
65:
66:  lea  window_defs,a0
67:  jsr  openwindow(a6)
68:  move.l  d0>window_hd
69:  beq  no_window
70:
71:  ----- view- und rastport ermitteln -----
72:
73:  move.l  screen_hd,a0
74:  add.l  #44,a0
75:  move.l  a0,viewport
76:  add.l  #40,a0
77:  move.l  a0,rastport
78:
79:  ----- bitplane-struktur umbauen -----
80:
81:  add.l  #108,a0      ; zeiger auf erste plane
82:  moveq  #4,d7        ; 5 planes sind vorhanden
83:
84:  free_plane_memory:
85:
86:  move.l  execbase,a6      ; speicher der einzelnen
87:  move.l  (a0)+,a1         ; planes freigeben
88:  move.l  #10240,d0
```

Listing: Fraktalgenerator

```
89:  move.l  a0,-(sp)
90:  jsr  freemem(a6)
91:  move.l  (sp)+,a0
92:  dbra  d7,free_plane_memory
93:
94:  move.l  #51200,d0      ; zusammenhaengenden
95:  move.l  #510002,d1     ; speicher fuer planes
96:  jsr  allocmem(a6)      ; holen
97:
98:  move.l  screen_hd,a0
99:  add.l  #192,a0
100:  moveq  #4,d7
101:
102:  set_new_pointers:
103:
104:  move.l  d0,(a0)+      ; adressen in bitplane-
105:  add.l  #10240,d0      ; struktur eintragen
106:  dbra  d7,set_new_pointers
107:
108:  move.l  intbase,a6      ; bildschirm neu aufbauen
109:  jsr  remakedisplay(a6)
110:
111:  ----- kommando-window oeffnen -----
112:
113:  jsr  wbenchtofront(a6)
114:
115:  lea  wbind_defs,a0      ; fenster oeffnen
116:  jsr  openwindow(a6)
117:  move.l  d0>wbind_hd
118:  beq  no_wbind          ; alles ok
119:                          ; nein => fehler
120:
121:  clr  pic_on_screen      ; flagge runter
122:  bsr  bin_to_asc         ; koordinaten nach ascii
123:
124:  wait_for_message:
125:
126:  move.l  execbase,a6
127:  move.l  wbind_hd,a0
128:  move.l  #6(a0),a0
129:  jsr  getmsg(a6)          ; message holen
130:  tst.l  d0               ; war was ?
131:  beq.s  wait_for_message ; nein => weiter warten
132:  move.l  d0,a0
133:  move  22(a0),d1          ; ereignis nach d1
134:  cmp  #540,d1            ; gadget angeklickt ?
135:  beq.s  gadget_up         ; ja => runter
136:  cmp  #5200,d1           ; close-gadget ?
137:  bne.s  wait_for_message ; nein => wieder rauf
138:
139:  ----- screen und windows wieder schliessen -----
140:
141:  move.l  intbase,a6
142:  move.l  wbind_hd,a0
143:  jsr  closewindow(a6)
144:
145:  no_wbind:
146:
147:  move.l  window_hd,a0
148:  jsr  closewindow(a6)
149:
150:  no_window:
151:
152:  move.l  screen_hd,a0
153:  jsr  closescreen(a6)
154:
155:  ----- libraries schliessen -----
156:
157:  no_screen:
158:
159:  move.l  execbase,a6
160:  move.l  gfxbase,a1
161:  jsr  closelibrary(a6)
162:
163:  no_gfx:
164:
165:  move.l  intbase,a1
166:  jsr  closelibrary(a6)
167:
168:  no_int:
169:
170:  move.l  dosbase,a1
171:  jsr  closelibrary(a6)
172:
173:  ----- programmende -----
174:
175:  no_dos:
176:
177:  rts
178:
179:  ----- gadget-kommandos ausfuehren -----
180:
181:  gadget_up:
182:
183:  move.l  28(a0),a0
184:  move  38(a0),d1
185:  cmp  #7,d1              ; fraktal berechnen ?
186:  beq  make_fractal       ; ja => dann los
187:  cmp  #8,d1              ; ausschnitt bestimmen ?
188:  beq  new_part            ; ja => auf gehts
```

Listing: Fraktalgenerator

### PUBLIC DOMAIN!!!

### Alles, was das Herz begehrt!!!

Jede 3,5"-Diskette, 135 TPI, 2DD  
ab 50 Stck. je Disk. nur DM 2,20  
ab 100 Stck. je Disk. nur DM 2, —  
nur DM 1,90

Jede 5,25"-Diskette  
ab 50 Stck. je Disk. nur DM 1,30  
ab 100 Stck. je Disk. nur DM 1,20  
nur DM 1,10

Wenn Sie uns Ihre Leerdisketten schicken möchten, berechnen wir je Diskette DM 0,80 als Kopiergebühr!!!

Wir haben über 3000 Disketten auf Lager, stets die aktuellsten Serien. Fordern Sie unsere 4 deutschen Katalogdisketten gegen DM 6, — in Briefmarken an!!! Es lohnt sich!!!

Übrigens liefern wir Ihre Bestellung spätestens einen Tag nach Eingang aus, meist noch am gleichen Tag!!! Und daß wir die Disketten prüfen, ist für uns selbstverständlich!!!

VIRUS-WARNER bei uns nur DM 29,95 inkl. Software!!!

Wir liefern auch kommerzielle Software und Hardware zu guten Konditionen. Und auch schnell!!! Rufen Sie an, und erfragen Sie unsere Preise!!!

**MVC** Musik Video Computer · Tel./Btx: 023 82/25 03  
Hammer Str. 103 · 4730 Ahlen

## Listing

```

188: cmp      #9,d1          ; polieren ?
189: beq      smooth
190: cmp      #10,d1

191: beq      save_picture    ; bild speichern
192: bra      wait_for_message ; wieder in die schleife
193:
194: #---- apfelmaennchen berechnen -----
195:
196: make_fractal:
197:
198: move     gdinfo_05+30,max_iter
199: bsr      check_entry      ; eingaben pruefen
200: bsr      asc_to_bin       ; koordinaten wandeln
201: bsr      fractal_to_front ; bildschirm nach vorn
202: move.l   screen_hd,a2
203: move.l   192(a2),a2
204: move.l   a2,-(sp)
205: move     #12799,d7       ; a2 auf stack sichern
206:                          ; bildschirm loeschen
207: clear_screen:
208:
209: clr.l    (a2)+           ; 12800 lang-worte
210: dbra     d7,clear_screen
211:
212: move.l   (sp)+,a2
213: bsr      make_table      ; plane-zeiger reparieren
214:                          ; koordinatentab. aufbauen
215: lea      table_q,a6      ; a6: zeiger ver.-tabelle
216:
217: move     #255,d1         ; screen hat 256 zeilen
218:
219: y_loop:
220:
221: move     d1,-(sp)
222: lea      table_p,a5
223: bsr      iteration
224: move     d0,store_00
225:
226: move     (sp)+,d1
227: move     #19,d0
228:
229: x_loop:
230:
231: movem    d0/d1,-(sp)
232: add.l    #32,a5
233: bsr      iteration
234: move     d0,store_08
235: cmp      store_00,d0
236: bne.s    erstes_byte_kaputt
237: move     #$ff00,d1
238: bsr      plot
239: bra      erstes_byte_okay
240:
241: erstes_byte_kaputt:
242:
243: sub.l    #16,a5
244: bsr      iteration
245: move     d0,store_04
246: cmp      store_00,d0
247: bne.s    erstes_nibble_kaputt
248: move     #$f000,d1
249: bsr      plot
250: bra      erstes_nibble_okay
251:
252: erstes_nibble_kaputt:
253:
254: move     store_00,d0
255: move     #$8000,d1
256: bsr      plot

257: sub.l    #12,a5
258: bsr      iteration
259: move     #$4000,d1
260: bsr      plot
261: addq.l   #4,a5
262: bsr      iteration
263: move     #$2000,d1
264: bsr      plot
265: addq.l   #4,a5
266: bsr      iteration
267: move     #$1000,d1
268: bsr      plot
269: addq.l   #4,a5
270: move     store_04,d0
271:
272: erstes_nibble_okay:
273:
274: cmp      store_08,d0
275: bne.s    zweites_nibble_kaputt
276: move     #$0f00,d1
277: bsr      plot
278: add.l    #16,a5
279: bra      erstes_byte_okay
280:
281: zweites_nibble_kaputt:
282:
283: move     store_04,d0
284: move     #$0800,d1
285: bsr      plot
286: addq.l   #4,a5
287: bsr      iteration
288: move     #$0400,d1
289: bsr      plot
290: addq.l   #4,a5
291: bsr      iteration
292: move     #$0200,d1
293: bsr      plot
294: addq.l   #4,a5
295: bsr      iteration
296: move     #$0100,d1
297: bsr      plot
298: addq.l   #4,a5
299:
300: erstes_byte_okay:
301:
302: add.l    #32,a5
303: bsr      iteration
304: move     d0,store_10
305: cmp      store_08,d0
306: bne.s    zweites_byte_kaputt
307: move     #$00ff,d1
308: bsr      plot
309: bra      zweites_byte_okay

```

Listing: Fraktalgenerator

# DONAU-SOFT

24 h-Schnellversand

Ihr Amiga-PD-Partner

● ab 2,50 DM ●

Alle gängigen Serien sind  
lieferbar

Einzeldisk 4,50 DM  
ab 10 Disk 4,— DM  
ab 50 Disk 3,50 DM  
ab 100 Disk 3,30 DM  
ab 200 Disk 3,— DM  
bei Serienabnahme: ab 2,50 DM

Preise incl. 3,5" DD-Disks

— Mit Qualitätsgarantie —

Wir kopieren nur mit doppeltem Verify.

Alle Disks sind:

— 100 % Virus- und Error frei  
— etikettiert.

Leerdisketten 3,5" 2DD von

Sentinel ab 1,25 DM  
Sony ab 1,70 DM

3 ausführliche Katalogdisketten  
mit Kurzbeschreibung aller  
Programme gegen 10,— DM  
(V-Scheck/Briefmarken) anfordern!

gratis zu unseren Katalogen:  
Viruskiller, CLJ-Wizard + Turbo Backup

Das große Amiga-PD-Handbuch  
Band I-IV + alle 42 Disks  
+ 3 Katalogdisketten  
(Einzelpreise erfragen) **325,—**

Pakete für Einsteiger und Anwender  
(jeweils 10 Disketten)

Einsteiger 1,2; Spiele 1,2,3;  
Sound; Grafik; Modula II  
jedes Einzelpaket 35,— DM  
3 Pakete nach Wahl nur 99,— DM

Floppy 3 1/2" Int. 155,— DM  
Floppy 3 1/2" ext. } abschaltbar 209,— DM  
Floppy 5 1/4" ext. } mit allen Extras 269,— DM

+ DM 5,— bei Vorkasse, + DM 8,— bei Nachnahme  
Ausland: + DM 10,— (nur Vorkasse)

## MAIK HAUER

Postfach 1401 8858 Neuburg Fax: 08431/49798  
Tel.: 08431/49798 (bis 22 Uhr) BKK \*Donau-Soft \*



```

310:
311: zweites_byte_kaputt:
312:
313: sub.l #16,a5
314: bsr iteration
315: move d0,store_0c
316: cmp store_08,d0
317: bne.s drittes_nibble_kaputt
318: move #00f0,d1
319: bsr plot
320: bra drittes_nibble_okay
321:
322: drittes_nibble_kaputt:
323:
324: move store_08,d0
325: move #0080,d1
326: bsr plot
327: sub.l #12,a5
328: bsr iteration
329: move #0040,d1
330: bsr plot
331: addq.l #4,a5
332: bsr iteration
333: move #0020,d1
334: bsr plot
335: addq.l #4,a5
336: bsr iteration
337: move #0010,d1
338: bsr plot
339: addq.l #4,a5
340:
341: drittes_nibble_okay:
342:
343: move store_0c,d0
344: cmp store_10,d0
345: bne.s viertes_nibble_kaputt
346: move #000f,d1
347: bsr plot
348: add.l #16,a5
349: bra zweites_byte_okay
350:
351: viertes_nibble_kaputt:
352:
353: move store_0c,d0
354: move #0008,d1
355: bsr plot
356: addq.l #4,a5
357: bsr iteration
358: move #0004,d1
359: bsr plot
360: addq.l #4,a5
361: bsr iteration
362: move #0002,d1
363: bsr plot
364: addq.l #4,a5
365: bsr iteration
366: move #0001,d1
367: bsr plot
368: addq.l #4,a5
369:
370: zweites_byte_okay:
371:
372: move store_10,store_00
373: addq.l #2,a2
374: movem (sp)+,d0/d1
375: dbra d0,x_loop
376: btst #6,$ffe001
377: beq.s exit_fractal
378: addq.l #4,a6
379: dbra d1,y_loop
380:
381: exit_fractal:
382:
383: move #-1,pic_on_screen
384: bsr wb_to_front
385: bra wait_for_message
386:
387: *---- tabelle mit koordinatenwerten aufbauen ----*
388:
389: make_table:
390:
391: move.l realfr,d0 ; differenz der anfangs-
392: move.l realto,d1 ; koordinaten bilden
393: sub.l d0,d1
394: move #320,d7 ; durch 320 teilen
395: bsr divide
396:
397: lea table_p,a0
398: move #319,d7 ; 320 werte eintragen
399:
400: p_loop:
401:
402: move.l d0,{a0}+ ; koordinaten jedes bild-
403: add.l d1,d0 ; schirmpunktes in tabel.
404: dbra d7,p_loop
405:
406: move.l imagfr,d0 ; abstand der imaginaeren
407: move.l imagto,d1 ; grenzen bestimmen
408: sub.l d0,d1
409: move #256,d7 ; durch 256 teilen
410: bsr divide
411:
412: lea table_q,a0
413: move #255,d7 ; 256 werte
414:
415: q_loop:
416:
417: move.l d0,{a0}+ ; jetzt koordinaten von
418: add.l d1,d0 ; zeilen eintragen
419: dbra d7,q_loop
420: rts
421:
422: *---- division (32 bit) ----*
423:
424: divide:
425:
426: move.l d1,d2
427: clr d1
428: swap d1
429: divu d7,d1
430: move.l d1,d3

```

Listing: Fraktalgenerator

```

431: swap d1
432: move d2,d3
433: divu d7,d3
434: move d3,d1
435:
436:
437: *---- iterationsroutine ----*
438:
439: iteration:
440:
441: sub.l a0,a0 ; x
442: sub.l a1,a1 ; y
443: moveq #0,d5 ; zaehler f. iterationen
444:
445: iteration_loop:
446:
447: move.l a1,d0 ; ist y negativ ?
448: btst #31,d0
449: beq.s y_positive
450: neg.l d0 ; => vorzeichen umdrehen
451:
452: y_positive:
453:
454: move.l d0,d1
455:
456: move d1,d2
457: swap d1
458: mulu d1,d2
459: moveq #0,d3
460: lsl.l #1,d2
461: roxl.l #1,d3
462: mulu d1,d1
463: moveq #0,d0
464: move d2,d0
465: swap d0
466: clr d2
467: add.l d3,d2
468: add.l d2,d1
469: move.l d1,d2
470: swap d2
471: and #f800,d2
472: bne overflow
473: rol.l #4,d1
474: and.b #f0,d1
475: rol.l #4,d0
476: and.b #f0f,d0
477: or.b d0,d1
478: move.l d1,d6
479:
480: move.l a0,d0 ; ist x negativ ?
481: btst #31,d0
482: beq.s x_positive
483: neg.l d0 ; => vorzeichen umdrehen
484:
485: x_positive:
486:
487: move.l d0,d1
488: move d1,d2
489: swap d1
490: mulu d1,d2
491: moveq #0,d3
492: lsl.l #1,d2
493: roxl.l #1,d3
494: mulu d1,d1
495: moveq #0,d0
496: move d2,d0
497: swap d0
498: clr d2
499: swap d2
500: add.l d3,d2
501: add.l d2,d1
502: move.l d1,d2
503: swap d2
504: and #f800,d2
505: bne overflow
506: rol.l #4,d1
507: and.b #f0,d1
508: rol.l #4,d0
509: and.b #f0f,d0
510: or.b d0,d1
511: move.l d1,d7
512:
513: move.l d6,d0 ; xx+yy>8 ?
514: add.l d7,d0
515: bcs overflow ; => abbruch
516: bmi overflow
517:
518: move.l a1,d0 ; y nach d0
519: move.l a0,d1 ; x nach d1
520: clr d4 ; vorzeichenflag loeschen
521:
522: move.l d0,d2 ; y zusaetzlich nach d2
523: btst #31,d2 ; ist y negativ ?
524: beq.s y_not_negative
525: neg.l d2 ; => d0 und d2 negieren
526: addq #1,d4 ; vorzeichenflag plus 1
527:
528: y_not_negative:
529:
530: move.l d1,d3 ; x zusaetzlich nach d3
531: btst #31,d3 ; ist x negativ ?
532: beq.s x_not_negative
533: neg.l d3 ; => d1 und d3 negieren
534: neg.l d1
535: addq #1,d4 ; vorzeichenflag plus 1
536:
537: x_not_negative:
538:
539: move d4,prefix ; vorzeichen merken
540:
541: move.l d0,d2
542: swap d2
543: mulu d1,d2
544: move.l d1,d3
545: swap d3
546: mulu d0,d3
547: swap d0
548: swap d1
549: mulu d1,d0
550: moveq #0,d4
551: move d2,d4
552: swap d4

```

Listing: Fraktaigenerator

# Listing

```

553: clr      d2
554: swap    d2
555: moveq   #0,d1
556: move    d3,d1
557: swap    d1
558: clr     d3
559: swap    d3
560: add.l   d4,d1
561: addx.l  d2,d3
562: add.l   d3,d0
563: rol.l   #5,d0
564: and.b   #5e0,d0
565: rol.l   #5,d1
566: and.b   #51f,d1
567: or.b    d1,d0
568: move    prefix,d4      ; bit 0 im vz-flag ?
569: btst    #0,d4          ; => = oder y war negativ
570: beq.s   result_positive ; also ergebnis negieren
571: neg.l   d0
572:
573: result_positive:
574:
575: move.l   d0,a0
576: sub.l    (a6),d0
577: move.l   d0,a0
578: move.l   d6,d1
579: move.l   d7,d0
580: sub.l    d0,d1
581: sub.l    (a5),d1
582: move.l   d1,a1
583: addq     #1,d5          ; iterationszaehler plus 1
584: cmp      max_iter,d5    ; grenze erreicht ?
585: bne      iteration_loop ; => ende, sonst nochmal
586: move     #0,d5          ; farbe = fuer grenze

587: overflow:
588:
589:
590: move     d5,d0          ; farbe nach d0
591: and      #15,d0         ; nur werte von 0 bis 31
592: rts
593:
594: *--- plot-routine -----*
595:
596: plot:
597:
598: asl      #4,d0          ; farbwert mal 16
599: jmp      zero(pc,d0.w)  ; in entsprechende routine
600:
601: zero:
602:
603: rts      ; farbe 0 (hier nix tun)
604: blk.b    14,0
605:
606: or       d1,0(a2)       ; 1
607: rts
608: blk.b    10,0
609:
610: or       d1,10240(a2)   ; 2
611: rts
612: blk.b    10,0
613:
614: or       d1,0(a2)       ; 3
615: or       d1,10240(a2)
616: rts
617: blk.b    6,0
618:
619: or       d1,20480(a2)   ; 4
620: rts
621: blk.b    10,0
622:
623: or       d1,0(a2)       ; 5
624: or       d1,20480(a2)
625: rts
626: blk.b    6,0
627:
628: or       d1,10240(a2)   ; 6
629: or       d1,20480(a2)
630: rts
631: blk.b    6,0
632:
633: or       d1,0(a2)       ; 7
634: or       d1,10240(a2)
635: or       d1,20480(a2)
636: rts
637: blk.b    2,0
638:
639: or       d1,30720(a2)   ; 8
640: rts
641: blk.b    10,0
642:
643: or       d1,0(a2)       ; 9
644: or       d1,30720(a2)
645: rts
646: blk.b    6,0
647:
648: or       d1,10240(a2)   ; 10
649: or       d1,30720(a2)
650: rts
651: blk.b    6,0
652:
653: or       d1,0(a2)       ; 11
654: or       d1,10240(a2)
655: or       d1,30720(a2)
656: rts
657: blk.b    2,0
658:
659: or       d1,20480(a2)   ; 12
660: or       d1,30720(a2)
661: rts
662: blk.b    6,0
663:
664: or       d1,0(a2)       ; 13
665: or       d1,20480(a2)
666: or       d1,30720(a2)
667: rts
668: blk.b    2,0
669:
670: or       d1,10240(a2)   ; 14
671: or       d1,20480(a2)
672: or       d1,30720(a2)
673: rts
674: blk.b    2,0

```

Listing: Fraktalgenerator

```

675:
676: or       d1,0(a2)       ; 15
677: or       d1,10240(a2)
678: or       d1,20480(a2)
679: or       d1,30720(a2)
680: rts
681:
682: *--- bildschirmteile polieren -----*
683:
684: smooth:
685:
686: tst      pic_on_screen  ; nur wenn bild berechnet
687: bne.s    smoothing_possible
688:
689: bsr      flash_workbench ; sonst screen flashen
690: beq      wait_for_message
691:
692: smoothing_possible:
693:
694: bsr      get_part       ; ausschnitt bestimmen
695: bsr      erase_cross     ; fadenkreuz loeschen
696:
697: move.l   gfxbase,a6
698: move.l   rastport,a1
699: moveq    #0,d0
700: jsr      setdrmd(a6)
701:
702: bsr      mousekey_hold
703:
704: lea      table_p,a5     ; ausgeahnten koordinaten
705: moveq    #0,d0
706: move     x1,d0          ; entsprechende adresse
707: lsl.l    #2,d0          ; in tabelle ermitteln
708: add.l    d0,a5
709: move.l   a5,p_pointer   ; sichern
710: lea      table_q,a6
711: moveq    #0,d0
712: move     y1,d0
713: lsl.l    #2,d0
714: add.l    d0,a6
715:
716: move     y1,d1          ; erste zu rechnende zeile
717:
718: y_count:
719:
720: move.l   p_pointer,a5
721: move     x1,d0          ; erste spalte
722:
723: x_count:
724:
725: movem.l  d0/d1/a5/a6,-(sp) ; auf den stack
726: movem    d0/d1,-(sp)      ; dito
727: jsr      iteration       ; iteration durchfuehren
728: move.l   gfxbase,a6
729: move.l   rastport,a1
730: jsr      setapen(a6)     ; farbe setzen
731: movem    (sp)+,d0/d1
732: jsr      writapixel(a6)  ; und dann den punkt
733: movem.l  (sp)+,d0/d1/a5/a6
734: addq     #4,a5           ; x-zeiger erhoehen
735: addq     #1,d0
736: cmp      x2,d0          ; ende der zeile ?
737: ble.s    x_count        ; nein => weitermachen
738: btst     #6,$bfe001     ; bei maustaste abbrechen
739: beq.s    exit_smoothing
740: addq     #4,a6           ; y-zeiger auch plus 1
741: addq     #1,d1          ; naechste zeile
742: cmp      y2,d1          ; alles fertig ?
743: ble.s    y_count        ; nein => schleife
744:
745: exit_smoothing:
746:
747: bsr      wb_to_front
748: bra      wait_for_message
749:
750: *--- neuen ausschnitt bestimmen -----*
751:
752: new_part:
753:
754: bsr      get_part       ; Ausschnitt waehlen
755: bsr      make_coordinates ; koordinaten bestimmen
756: bsr      bin_to_asc     ; festkomma in ascii
757: bsr      wb_to_front    ; workbench nach vorn
758: bsr      erase_cross     ; fadenkreuze loeschen
759: bra      wait_for_message
760:
761: make_coordinates:
762:
763: lea      table_p,a0     ; koordinate des
764: move     x1,d0          ; punktes neuer anfang
765: lsl      #2,d0
766: move.l   (a0,d0.w),realfr ; real-anfang
767:
768: move     x2,d0
769: lsl      #2,d0
770: move.l   (a0,d0.w),realto ; real-ende
771:
772: lea      table_q,a0
773: move     y1,d0
774: lsl      #2,d0
775: move.l   (a0,d0.w),imagfr ; imaginuer-anfang
776:
777: move     y2,d0
778: lsl      #2,d0
779: move.l   (a0,d0.w),imagto ; imaginuer-ende
780:
781: rts
782:
783: *--- ausschnitt waehlen -----*
784:
785: get_part:
786:
787: tst      pic_on_screen  ; schon bild gerechnet ?
788: bne.s    picture_there  ; ja => also weitermachen
789:
790: bsr      flash_workbench ; bildschirm aufblitzen
791: addq     #4,sp
792: bra      wait_for_message ; steckpointer korrig.
793:                                     ; heim ins menue
794:
795: picture_there:
796:
797: move.l   gfxbase,a6

```

Listing: Fraktalgenerator



```

797: move.l   rastport,a1
798: moveq    #2,d0
799: jsr      setdrmd(a6)
800:
801: bsr      fractal_to_front ; screen nach vorn holen
802: move.l   window_hd,a5    ; zeiger auf das fenster
803: bsr      part_sub         ; koordinaten links oben
804: move     d6,y1            ; bestimmen
805: move     d7,x1
806:
807: bsr      mousekey_hold    ; warten, bis taste weg
808:
809: bsr      part_sub         ; koordinaten rechts u.
810: move     d6,y2            ; bestimmen
811: move     d7,x2
812:
813: move     x1,d0            ; anfangsordinate klein-
814: cmp      x2,d0            ; ner endkoordinate ?
815: blt.s    x1_lt_x2
816:
817: move     x2,x1            ; sonst rumdrehen
818: move     d0,x2
819:
820: x1_lt_x2:
821:
822: move     y1,d0            ; dito fuer y-werte
823: cmp      y2,d0
824: blt.s    y1_lt_y2
825:
826: move     y2,y1
827: move     d0,y2
828:
829: y1_lt_y2:
830:
831: rts
832:
833: part_sub:
834:
835: move     12(a5),d6        ; maus y-koordinate
836: move     14(a5),d7        ; x
837: bsr      draw_cross       ; fadenkreuz zeichnen
838:
839: mouse_not_moved:
840:
841: cmp      14(a5),d7        ; koordinaten noch gleich
842: bne.s    mouse_moved      ; => maus nicht bewegt
843: cmp      12(a5),d6
844: bne.s    mouse_moved
845: btst     #6,$b7e001       ; linke maustaste
846: bne.s    mouse_not_moved  ; => schleife beenden
847:
848: bsr      draw_cross       ; altes kreuz loeschen
849: move     12(a5),d6        ; aktuelle koordinaten
850: move     14(a5),d7        ; sichern
851:
852: bra      draw_cross       ; fadenkreuz malen
853:
854: mouse_moved:
855: bsr      draw_cross       ; kreuz loeschen
856: bsr.s    part_sub        ; wieder in die schleife
857:
858: *---- fadenkreuz zeichnen -----*
859:
860: draw_cross:
861:
862: move.l   gfxbase,a5       ; horizontale linie
863: move.l   rastport,a1
864: moveq    #0,d0
865: move     d6,d1
866: jsr      move(a6)
867: move     #319,d0
868: move     d6,d1
869: jsr      draw(a6)
870:
871: move     d7,d0            ; jetzt die vertikale
872: moveq    #0,d1
873: jsr      move(a6)
874: move     d7,d0
875: move     #255,d1
876: jmp      draw(a6)
877:
878: erase_cross:
879:
880: move     x1,d7
881: move     y1,d6
882: bsr.s    draw_cross
883: move     x2,d7
884: move     y2,d6
885: bsr.s    draw_cross
886:
887: *---- eingaben ueberpruefen -----*
888:
889: check_entry:
890:
891: rts
892:
893: lea      gdbuff_01,a0
894: bsr      ce_sub
895:
896: lea      gdbuff_02,a0
897: bsr      ce_sub
898:
899: lea      gdbuff_03,a0
900: bsr      ce_sub
901:
902: lea      gdbuff_04,a0
903:
904: ce_sub:
905:
906: move.b   (a0)+,d0         ; vorzeichen nach d0
907: cmp.b    #"-",d0          ; minus ?
908: beq.s    prefix_ok        ; ja => alles klar
909: cmp.b    #"+",d0          ; plus ?
910: beq.s    prefix_ok        ; ja => auch in ordnung
911:
912: moveq    #9,d7            ; sonst alles 1 nach
913: bsr      move_it          ; hinten
914: move.b   #"+",-1(a0)      ; plus einsetzen
915:
916: prefix_ok:
917:
918: cmp.b    #".",a0         ; zeichen 2 nicht punkt ?

```

Listing: Fraktalgenerator

```

919: bne.s    digit_found      ; ja => dann weiter
920:
921: moveq    #8,d7            ; sonst auch hier zurueck
922: bsr      move_it          ;
923: move.b   #0",a0          ; null an zweite stelle
924:
925: digit_found:
926:
927: move.l   intbase,a6       ; gadgets aktualisieren
928: lea      gadget_01,a0
929: move.l   wvwind_hd,a1
930: sub.l    a2,a2
931: jmp      refreshgadgets(a6)
932:
933: move_it:
934:
935: move.l   a0,a1
936: addq.l   #8,a1
937:
938: move_loop:
939:
940: move.b   (a1),1(a1)
941: subq.l   #1,a1
942: dbra     d7,move_loop
943: clr.b    9(a0)
944: rts
945:
946: *---- koordinaten von binaer nach ascii wandeln -----*
947:
948: bin_to_asc:
949:
950: move.l   realfr,d0
951: lea      gdbuff_01,a0
952: bsr      ba_sub
953:
954: move.l   realto,d0
955: lea      gdbuff_02,a0
956: bsr      ba_sub
957:
958: move.l   imagfr,d0
959: lea      gdbuff_03,a0
960: bsr      ba_sub
961:
962: move.l   imagto,d0
963: lea      gdbuff_04,a0
964: bsr      ba_sub
965:
966: move.l   intbase,a6
967: lea      gadget_01,a0
968: move.l   wvwind_hd,a1
969: sub.l    a2,a2
970: jmp      refreshgadgets(a6)
971:
972: ba_sub:
973:
974: moveq    #8,d7
975: move.b   #"+",a0          ; plus an anfang setzen
976: tst.l    d0               ; koordinatenwert testen
977: bge.s    positive        ; positiv => weiter
978: move.b   #"-",a0          ; sonst minus einsetzen
979: neg.l    d0               ; und wert negieren
980:
981: positive:
982:
983: addq.l   #1,a0            ; naechste stelle
984: lea      table,a1         ; ptr auf wertigkeiten
985:
986: ba_loop_01:
987:
988: move.l   (a1)+,d2         ; wertigkeit aus tabelle
989: move.b   #48,d6           ; ascii "0" nach d6
990:
991: ba_loop_02:
992:
993: sub.l    d2,d0            ; wertigkeit abziehen
994: bmi.s    ba_exit         ; erg. negativ => weiter
995: addq.b   #1,d6           ; ascii-wert plus eins
996: cmp.b    #57,d6          ; "9" schon erreicht ?
997: bne.s    ba_loop_02      ; nein => weiter
998: sub.l    d2,d0
999:
1000: ba_exit:
1001:
1002: move.b   d6,(a0)+         ; ascii-wert eintragen
1003: add.l    d2,d0            ; wert wieder addieren
1004: cmp      #8,d7           ; stelle acht bearbeitet?
1005: bne.s    no_point        ; ja => punkt setzen
1006: move.b   #".",a0+
1007:
1008: no_point:
1009:
1010: dbra     d7,ba_loop_01    ; wieder in die schleife
1011: clr.b    -(a0)            ; letzte stelle auf null
1012: cmp.b    #9",-(a0)        ; letzte stelle eine 9 ?
1013: bne.s    no_round        ; nein => nicht runden
1014:
1015: round_loop:
1016:
1017: move.b   #0",a0          ; sonst gegen 0 tauschen
1018:
1019: point:
1020:
1021: cmp.b    #9",-(a0)        ; naechste stelle auch 9?
1022: beq.s    round_loop      ; ja => weiter runden
1023: cmp.b    #".",a0         ; punkt ueberspringen
1024: beq.s    point
1025: addq.b   #1,a0
1026:
1027: no_round:
1028:
1029: rts
1030:
1031: *---- jetzt in der umgekehrten richtung wandeln -----*
1032:
1033: asc_to_bin:
1034:
1035: lea      gdbuff_01,a0
1036: bsr      ab_sub
1037: move.l   d0,realfr
1038:
1039: lea      gdbuff_02,a0
1040: bsr      ab_sub
1041: move.l   d0,realto
1042:

```

Listing: Fraktalgenerator

# Listing

```

1043: lea      gdbuff_03,a0
1044: bsr      ab_sub
1045: move.l   d0,imagfr
1046:
1047: lea      gdbuff_04,a0
1048: bsr      ab_sub
1049:
1050: move.l   d0,imagto
1051: rts
1052:
1053: ab_sub:
1054:
1055: move.b   (a0)+,d5      ; vorzeichen merken
1056: cmp.b    #48,d5        ; zeichen < asc "0" ?
1057: blt.s    first_no_digit ; => wird vorzeichen sein
1058:
1059: move.b   #"+",d5        ; sonst plus annehmen
1060: subq.l   #1,a0          ; pointer minus 1
1061:
1062: first_no_digit:
1063: lea      table,a1      ; ptr auf wertetab.
1064: moveq    #0,d0          ; zahl erstmal auf null
1065: moveq    #8,d7
1066:
1067: ab_loop_01:
1068:
1069: move.l   (a1)+,d2        ; wert aus tabelle holen
1070: moveq    #0,d6
1071: move.b   (a0)+,d6        ; eine stelle nach d6
1072: sub      #48,d6         ; 48 (asc "0") abziehen
1073: beq.s    digit_is_zero  ; ergebnis <= 0 ?
1074: bmi.s    digit_is_zero  ; => nichts tun
1075:
1076: ab_loop_02:
1077:
1078: add.l    d2,d0          ; wert der stelle addieren
1079: subq     #1,d6          ; das ganze d6 mal
1080: bne.s    ab_loop_02 ;bgt ?
1081:
1082: digit_is_zero:
1083:
1084: cmp      #8,d7          ; stelle 8 an der reihe ?
1085: bne.s    digit_ok       ; nein => dann alles ok
1086: move.b   (a0)+,d6       ; naechste stelle lesen
1087:
1088: digit_ok:
1089:
1090: dbra     d7,ab_loop_01  ; naechste stelle
1091: cmp.b    #"+",d5        ; vorzeichen positiv ?
1092: beq.s    value_is_positive ; => wert so lassen
1093:
1094: neg.l     d0            ; ansonsten negieren
1095:
1096: value_is_positive:
1097:
1098: rts
1099:
1100: *---- screens umschalten -----*
1101:
1102: wb_to_front:
1103:
1104: move.l   intbase,a6
1105: jmp      wbenchtofront(a6)
1106:
1107: fractal_to_front:
1108:
1109: move.l   intbase,a6
1110: move.l   screen_hd,a0
1111: jmp      screenofront(a6)
1112:
1113: *---- workbench aufblitzen lassen -----*
1114:
1115: flash_workbench:
1116:
1117: move.l   intbase,a6
1118: sub.l    a0,a0
1119: jmp      displaybeep(a6)
1120:
1121: *---- warten, bis linke maustaste losgelassen -----*
1122:
1123: mousekey_hold:
1124:
1125: btst     #6,$bfe001
1126: beq.s    mousekey_hold
1127: rts
1128:
1129: *---- bild auf diskette abspeichern -----*
1130:
1131: save_picture:
1132:
1133: tst.b    gdbuff_06
1134: bne.s    filename_there
1135:
1136: bsr      flash_workbench
1137: bra      wait_for_message
1138:
1139: filename_there:
1140:
1141: move.l   dosbase,a6
1142: move.l   #gdbuff_06,d1
1143: move.l   #1006,d2
1144: jsr      open(a6)
1145: move.l   d0,file_hd
1146: beq.s    file_error
1147:
1148: move.l   #iff_header,d2
1149: move.l   #48,d3
1150: bsr      write_data
1151:
1152: move.l   screen_hd,a4
1153: add.l    #192,a4
1154: moveq    #0,d4          ; zeile
1155:
1156: loop_y:
1157:
1158: moveq    #0,d5          ; plane
1159:
1160: loop_p:
1161:
1162: move.l   (a4,d5.w),d2
1163: add.l    d4,d2
1164: moveq    #40,d3
1165: bsr      write_data

```

Listing: Fraktalgenerator

```

1166: addq     #4,d5
1167: cmp      #16,d5
1168: blt.s    loop_p
1169:
1170: add.l    #40,d4
1171: cmp.l    #10240,d4
1172: blt.s    loop_y
1173:
1174: close_file:
1175:
1176: move.l   dosbase,a6
1177: move.l   file_hd,d1
1178: jsr      close(a6)
1179: beq      file_error
1180:
1181:
1182: bra      wait_for_message
1183:
1184: write_data:
1185:
1186: move.l   file_hd,d1
1187: jsr      write(a6)
1188: cmp.l    #-1,d0
1189: bne.s    write_okay
1190:
1191: addq.l   #4,sp
1192: bsr      flash_workbench
1193: bra.s    close_file
1194:
1195: write_okay:
1196:
1197: rts
1198:
1199: file_error:
1200:
1201: bsr      flash_workbench
1202: bra      wait_for_message
1203:
1204: iff_header:
1205:
1206: dc.b     "FORM"
1207: dc.l     41000
1208: dc.b     "ILBMFORM"
1209: dc.l     20
1210: dc.w     320,256,0,0
1211: dc.b     4,0,0,0
1212: dc.w     0
1213: dc.b     10,11
1214: dc.w     320,256
1215: dc.b     "BODY"
1216: dc.l     40960
1217:
1218: file_hd:
1219:
1220: dc.l     0
1221:
1222: *---- gadget-strukturen -----*
1223:
1224: gadget_01:
1225: dc.l     gadget_02
1226: dc.w     150,20,136,10,0,3,4
1227: dc.l     gdbord_01,0,gdtext_01,0,gdinfo_01
1228: dc.w     1,0,0
1229: dc.l     0,gdbuff_01,0
1230: dc.w     0,11,0,0,0,0,0,0
1231: dc.l     0,0,0
1232: dc.b     16,0
1233: dc.w     1,0,1,0
1234: dc.w     -110,0
1235: dc.l     0,gdtptr_01,0
1236: dc.b     "Realität von",0,0
1237: dc.w     0,0
1238: dc.l     1,1,0,5
1239: dc.l     gdkord_01,0
1240: dc.w     -3,-3,137,-3,137,10,-3,10,-3,-3
1241:
1242: gadget_02:
1243: dc.l     gadget_03
1244: dc.w     340,20,136,10,0,3,4
1245: dc.l     gdbord_01,0,gdtext_02,0,gdinfo_02
1246: dc.w     2,0,0
1247: dc.l     gdbuff_02,0
1248: dc.w     0,11,0,0,0,0,0,0
1249: dc.l     0,0,0
1250:
1251: gdbuff_02:
1252: dc.b     16,0
1253: dc.b     1,0,1,0
1254: dc.w     -38,0
1255: dc.l     0,gdtptr_02,0
1256: dc.b     "bis",0
1257:
1258: gadget_03:
1259: dc.l     gadget_04
1260: dc.w     150,36,136,10,0,3,4
1261: dc.l     gdbord_01,0,gdtext_03,0,gdinfo_03
1262: dc.w     3,0,0
1263: dc.l     gdbuff_03,0
1264: dc.w     0,11,0,0,0,0,0,0
1265: dc.l     0,0,0
1266: dc.b     16,0
1267: dc.w     1,0,1,0
1268: dc.w     -110,0
1269: dc.l     0,gdtptr_03,0
1270: dc.b     "Imagteit von",0,0
1271:
1272: gadget_04:
1273: dc.l     gadget_05
1274: dc.w     340,36,136,10,0,3,4
1275: dc.l     gdbord_01,0,gdtext_02,0,gdinfo_04
1276: dc.w     4,0,0
1277: dc.l     gdbuff_04,0
1278: dc.w     0,11,0,0,0,0,0,0
1279: dc.l     0,0,0
1280: dc.b     16,0
1281:
1282: gdbuff_04:
1283: dc.b     16,0
1284: dc.b     1,0,1,0
1285: dc.w     -110,0
1286: dc.l     0,gdtptr_05,0

```

Listing: Fraktalgenerator



## Listing

```

1287: gdtptr_05: dc.b "Rechentiefe:",0,0
1288:
1289: gadget_06: dc.l gadget_07
1290: dc.w 40,103,280,10,0,3,4
1291: dc.l gbbord_06,0,gdtext_06,0,gdinfo_06
1292: dc.w 6,0,0
1293: gdinfo_06: dc.l gdbuff_06,0
1294: dc.w 0,35,0,0,0,0,0,0
1295: dc.l 0,0,0
1296: gdbuff_06: blk.b 40,0
1297: gdtext_06: dc.b 1,0,1,0
1298: dc.w 6,-13
1299: dc.l 0,gdtptr_06,0
1300: gdtptr_06: dc.b "Filename:",0
1301: gdbord_06: dc.w 0,0
1302: dc.b 1,1,0,5
1303: dc.l gdkord_06,0
1304: gdkord_06: dc.w -3,-3,284,-3,284,10,-3,10,-3,-3
1305:
1306: gadget_07: dc.l gadget_08
1307: dc.w 340,56,136,8,0,3,1
1308: dc.l gdbord_01,0,gdtext_07,0,0
1309: dc.w 7,0,0
1310: gdtext_07: dc.b 1,0,1,0
1311: dc.w 12,0
1312: dc.l 0,gdtptr_07,0
1313:
1314: gdtptr_07: dc.b "Bild berechnen",0,0
1315:
1316: gadget_08: dc.l gadget_09
1317: dc.w 340,71,136,8,0,3,1
1318: dc.l gdbord_01,0,gdtext_08,0,0
1319: dc.w 8,0,0
1320: gdtext_08: dc.b 1,0,1,0
1321: dc.w 28,0
1322: gdtptr_08: dc.l 0,gdtptr_08,0
1323: dc.b "AuschnTtt",0,0
1324:
1325: gadget_09: dc.l gadget_10
1326: dc.w 340,87,136,8,0,3,1
1327: dc.l gdbord_01,0,gdtext_09,0,0
1328: gdtext_09: dc.w 9,0,0
1329: dc.b 1,0,1,0
1330: dc.w 36,0
1331: gdtptr_09: dc.l 0,gdtptr_09,0
1332: dc.b "Polieren",0,0
1333:
1334: gadget_10: dc.l 0
1335: dc.w 340,103,136,8,0,3,1
1336: dc.l gdbord_01,0,gdtext_10,0,0
1337: gdtext_10: dc.w 10,0,0
1338: dc.b 1,0,1,0
1339: dc.w 12,0
1340: gdtptr_10: dc.l 0,gdtptr_10,0
1341: dc.b "Bild speichern",0,0
1342:
1343: -----
1344: dosname: dc.b "dos.library",0
1345: dosbase: dc.l 0
1346:
1347: intname: dc.b "intuition.library",0
1348: intbase: dc.l 0
1349:
1350: gfxbase: dc.b "graphics.library",0,0
1351: gfxname: dc.l 0
1352:
1353: viewport: dc.l 0
1354: rastport: dc.l 0
1355:
1356: screen_defs: dc.w 0,0,320,256,5
1357: dc.b 0,1
1358: dc.w 2,5
1359: dc.l 0,0,0,0
1360:
1361: window_defs: dc.w 0,0,320,256
1362: dc.b 0,1
1363: dc.l $400,$11800,0,0,0
1364: screen_hd: dc.l 0,0
1365: dc.w 320,256,320,256,15
1366: window_hd: dc.l 0
1367:
1368: wbind_defs: dc.w 40,40,560,120
1369: dc.b 0,1
1370: dc.l $240,$1100e,gadget_01,0
1371: wbind_name: dc.l 0,0
1372: dc.w 560,120,560,120,1
1373: wbind_hd: dc.b "FrakGen AmigaDOS V1.01",0,0
1374: dc.l 0
1375:
1376: realfr: dc.l $c8000000
1377: realto: dc.l $37ffffff
1378: imagfr: dc.l $c8000000
1379:
1380: imagto: dc.l $37ffffff
1381: max_iter: dc.w 0
1382:
1383: pic_on_screen: dc.w 0
1384: prefix: dc.w 0
1385:
1386: table_p: blk.b 1280,0
1387: table_q: blk.b 1024,0
1388: p_pointer: dc.l 0
1389:
1390: x1: dc.w 0
1391: y1: dc.w 0
1392: x2: dc.w 0
1393: y2: dc.w 0
1394:
1395: store_00: dc.w 0
1396: store_04: dc.w 0
1397: store_08: dc.w 0
1398: store_0c: dc.w 0
1399: store_10: dc.w 0
1400:
1401: table: dc.l $10000000/1
1402: dc.l $10000000/10
1403: dc.l $10000000/100
1404: dc.l $10000000/1000
1405: dc.l $10000000/10000
1406: dc.l $10000000/100000
1407: dc.l $10000000/1000000
1408: dc.l $10000000/10000000
1409: dc.l $10000000/100000000
1410:
1411: -----

```

Listing: Fraktalgenerator

## Filecards Amiga2000:

**ALF2 SCSI-Kontroller** (inkl. Software) mit  
**Seagate ST157N-0** 48MB 35ms **1299.--**  
**Seagate ST157N-1** 48MB 28ms **1399.--**  
**Seagate ST1096N** 83MB 24ms **1799.--**  
**Quantum ProDrive** 42MB 19ms **1749.--**

## Speichererweiterungen:

**Amiga2000 8MB, 2MB bestückt mit**  
**511000 100ns** (autokonfigurierend) **749.--**  
**dito 4MB bestückt** **1099.--**  
**dito 8MB bestückt** **1799.--**  
**Amiga500 2MB Ramkarte** **599.--**

mit FAT-AGNUS 1.8MB, mit BIG-AGNUS  
 volle 2MB (Chip/Fastram), intern

## 68030Karte GVP

**28MHz, mit AT-Kontroller** **2199.--**

## Software:

**Professional Page** **399.--**  
**X-CAD Professional** **799.--**  
**M2Amiga Compiler 3.3** **299.--**

Weitere Software auf Anfrage

Leerdisketten  
 NoName 2DD  
 10 Stück 13.--

**AURIGA Technologie** Eiselt Stefan  
 Mainaustr. 38 · 8000 München 60 · Tel.: 089/8203651  
 Info anfordern / Händleranfragen erwünscht



## Spitzen- Zubehör für AMIGA-PC's

**512K Ram für AMIGA-500** **159,50 DM**  
 mit 4 Stück 1 Mbit Chips !!!, Accu-Uhr, abschaltbar

**128K RAM-Karte für XT-Karte:** **198,00 DM**  
 kurze Karte mit 128K Ram. Somit haben Sie 640K DosRam  
 und können z.B. Ventura, F&A und D-Base-IV anwenden !!!

**XT-Multifunktions Karte:** **299,00 DM**  
 Multi-IO-Karte für XT-Karte mit 128K Ram, Ser., Par. & Uhr

**A2000 ALF2 Autoboot Filecards:**  
 20MB:888,-DM, 30MB:968,-DM, 50MB:1268,-DM

**A500 ALF2 Autoboot Festplatten:**  
 20MB:1068,-DM, 30 MB:1168,-DM, 40MB:1448,-DM

**File-Cards für XT-Karte / SideCar:**  
 30 MB / 60ms: 668,-DM, 50 MB / 40ms:948,-DM

Die XT-File-Cards können für AMIGA & XT partitioniert werden.  
 Lieferung incl. ausführlicher Soft- & Hardware Installationsanleitung.

**Fordern Sie ~~unser~~ kostenlose Preisliste**

**SKy-Ware, Postf.1225, 5358 Illnd Münstereifel**  
**Telefon: 02253/2667 (btx)**



# Gewußt wie!

## Die Seiten für Einsteiger und Fortgeschrittene

Ein neuer Monat, eine neue AMIGA DOS und damit auch neue 'Gewußt wie'-Seiten. Zuerst jedoch mal ein Lob an alle Einsender: Wir finden es prima, daß sehr viele Einsender unserem Aufruf gefolgt sind und uns ihr Wissen zur Weitergabe an die Leser übermittelt haben. Deshalb Danke an alle, die bisher mitgemacht haben und die noch mitmachen werden. Doch nun wieder zu unseren Kurztips.

### HardError-Disks weaternutzen

Ist es Ihnen auch schon einmal passiert, daß sich plötzlich ein HardError auf einer Ihrer Disketten befand? Bisher konnten Sie solche Disks getrost dem Mülleimer überlassen. Damit ist jetzt Schluß! Sie brauchen nur die Mountlist, die sich im "devs"-Ordner Ihrer Startdiskette befindet, entsprechend abzuändern.

Ein Beispiel: Sie haben auf einer Diskette mehrere HardErrors auf den Spuren 70-79 und Sie besitzen zwei Diskettenlaufwerke. Fügen Sie folgenden Eintrag in die Mountlist Ihrer Startdiskette ein (die Zeilennummern nicht mit eingeben!). Dafür begeben Sie sich ins CLI oder in die Shell und geben dort ein "ed devs:mountlist", gefolgt von einem Return. Schreiben Sie den untenstehenden ersten

Eintrag ans Ende der Mountlist, drücken Sie dann die Tasten <ESC> und <X>, gefolgt von einem Return.

```
Eintrag 1:
1 Krankenhaus: Device =
      trackdisk.device
2 Unit = 1
3 Flags = 1
4 Surfaces = 2
5 BlocksPerTrack = 11
6 Reserved = 2
7 Interleave = 0
8 LowCyl = 0;
      HighCyl = 69
9 Buffers = 20
10 BufMemType = 3
11 #
```

Der Wert LowCyl gibt die unterste zu formatierende Spur an (0), und HighCyl gibt die höchste zu formatierende Spur an (69). Normalerweise besitzt eine Disk die Spuren 0 bis 79, da aber 70 bis 79 defekt sind, können wir nur 0 bis 69 nutzen. Geben Sie nun im CLI/Shell den Befehl "mount Krankenhaus:" ein, um Ihr

neues logisches Device Krankenhaus: anzumelden. Der Wert Unit in Zeile 2 gibt an, daß das Laufwerk df1: das Device Krankenhaus: simuliert. Sie können nun die defekte Diskette in Drive df1: einlegen und selbige mit folgendem Befehl formatieren:

```
sys:system/format drive
Krankenhaus: name testdisk
noicons
```

Die Diskette wird nun formatiert und ist für Sie wieder nutzbar, jedoch mit einer geringeren Speicherkapazität. Sollten Sie mehrere Disketten mit diesem Trick formatieren, die verschiedene defekte Spuren haben, so müssen Sie vor dem Einlegen einer anderen vorher defekten Disk das bisher genutzte logische Device wieder abmelden (ist jedenfalls aus Speicherplatzgründen besser). Dies geht ganz einfach mit dem Befehl "assign remove Krankenhaus:".

Nun können Sie das andere Device anmelden (z.B. mount Krankenhaus2:) und die Disk verwenden. Sie brauchen also für jede mit anderen Low- und HighCyl-Werten formatierte Diskette einen eigenen Mountlist-Eintrag. Für CLI-Geübte ist dies sicherlich einfach zu nutzen, alle anderen sollten damit erst einmal etwas herumprobieren (nur unformatierte, also leere Disketten benutzen).

### FastFileSystem auf Disketten

Es wurde ja immer behauptet, man könne das FastFileSystem nur auf Festplatten und RamDisks anwenden. Stimmt aber nicht! Sie müssen nur folgenden Mountlist-Eintrag an die Mountlist Ihrer Startdiskette anfügen und schon kann's losgehen:

```
1 FFS-Drive: Device = tra
      ckdisk.device
2 FileSystem = 1:FastFileSy
      stem
3 Unit = 1
4 Flags = 1
5 Surfaces = 2
6 BlocksPerTrack = 11
7 Reserved = 2
8 Interleave = 0
9 LowCyl = 0; HighCyl = 79
10 Buffers = 20
11 GlobVec = -1
12 BufMemType = 3
13 Mount = 1
14 DosType = 0x444F5301
15 #
```

Melden Sie das neue logische Device mit "mount FFS-Drive:" an. Legen Sie nun eine leere, unformatierte Disk in Drive df1: und formatieren Sie diese mit folgendem Befehl:

```
sys:system/format drive FF
S-Drive: name ffstdisk ffs
noicons
```

Die Diskette bietet nun 5% mehr Speicherplatz und wird bedeutend schneller geladen. Am extremsten bemerkt man dies beim "dir"-Befehl. Wollen Sie mehrere Disks mit dem FFS benutzen, so beachten Sie bitte, daß immer erst das FFS-Drive-Device mit "mount FFS-Drive:" vor dem ersten Einsatz einer FFS-formatierten Diskette angemeldet werden muß. Des weiteren beachten Sie bitte, daß das FastFileSystem keinen Diskettenwechsel erkennt, da eine Festplatte oder RamDisk im allgemeinen nicht gewechselt wird. Hierfür verwenden Sie bitte den "disk-



change-Befehl, den Sie wie alle anderen Befehle im C-Verzeichnis Ihrer Diskette finden.

## Schnelle Headerfiles mit dem 'Lattice C'-Compiler

Was Aztec-Besitzer schon lange genießen, ist nun auch Lattice-Usern seit der Version 5.02 ihres Compilers vergönnt. Ich spreche von der Möglichkeit, Headerfiles vorzucompilieren (auch PreCompiling). Hiermit müssen die Include-Files nicht immer neu übersetzt werden, sondern nur der Quelltext des Hauptmoduls. Hierbei wird das File sozusagen vorcompiliert. Die Includefiles werden als Symboltabelle im Quad-Device gespeichert. Beim endgültigen Compilieren wird dieser Code wieder geladen. Die Include-Anweisungen im Quellprogramm müssen selbstverständlich wegfallen. Nehmen wir an, Ihr Programm benutzt nur die Headerfiles "dos.h" und "intuition/intuition.h". Um diese zu precompilieren, erstellen Sie sich eine Datei mit dem Namen 'headers.c' (Beispielnamen!), die folgenden Inhalt hat:

```
#include <dos.h> #include
<intuition/intuition.h>
```

Nun rufen Sie den Compiler mit folgender Anweisung auf: `lc1-ph headers.c`

Er erzeugt eine Datei mit dem Namen 'headers.q'. Wollen Sie nun Ihr Hauptprogramm compilieren, so rufen Sie den `lc1-Pass` folgendermaßen auf:

```
lc1-Hheaders.q Hauptprogramm.c
```

Zwischen der Option -H und `headers.q` darf kein Leerzeichen sein! Im normalen Programmcode dürfen natürlich nicht mehr die Include-Dateien aufgerufen werden, die bereits übersetzt wurden. Andere Includes dürfen natürlich verwendet werden.

## AutoRequest V1.1

"AReq" erzeugt einen Auto-Requester im aktuellen Window mit den vom Benutzer übergebenen Parametern für Body, linkes und rechtes Gadget. Der Quellcode wurde mit dem 'Lattice C'-Compiler V5.02 erstellt, dürfte jedoch auch dem Aztec-C-V3.6a-Compiler keine Probleme be-

reiten. Es existieren weiter zwei Make-Dateien, die ich aufgrund ihrer Einfachheit als Batch-Dateien erstellt habe. 'make' erzeugt die normale Version des Programms und 'make.res' erzeugt die residentfähige Version. Weiterhin habe ich eine Beispiel-Batch-Datei beigelegt, die die Anwendung des Befehls demonstriert. Sollten noch Fragen offen sein, so stehe ich gerne über die Redaktion der AMIGA DOS zur Verfügung.

Übrigens, dies ist die Version 1.1 des Programms. In dieser neuen Version habe ich die Argumentübergabe verändert und den Speicherbedarf des Programms auf Diskette reduziert. Bei der normalen Version hat dies immerhin 192 Bytes eingebracht und bei der residentfähigen fast 165. Die größte Speicherplatzersparnis brachte die Entfernung der Abfrage, ob die 'Intuition.Library' geöffnet werden konnte. Diese Abfrage halte ich sowieso für überflüssig, da sie immer vorhanden ist. Es ist zwar kein guter Programmierstil, aber der Zweck heiligt die Mittel.

Info:  
 Programm: 'AReq'  
 Filelänge: 3276 Bytes  
 (Die Version für Speichersparer!)  
 Filelänge: 3360 Bytes  
 (Diese Version kann resident gemacht werden (RESIDENT-BEFEHL))  
 Aufruf: 'AReq' Body LGadget  
 RGadget  
 Body = Titel  
 RGadget = Text für rechtes Gadget  
 LGadget = Text für linkes Gadget

Texte können auch Leerzeichen beinhalten, müssen dann aber in Anführungszeichen übergeben werden.

Rückgabe:  
 FEHLER: 10 bei falschem Aufruf  
 Sonstige: 5 wenn linkes Gadget gedrückt (WARN)

0 wenn rechtes Gadget gedrückt

Beispiel : siehe 'AReq.batch' (Thomas Knoth/jb)

## Große Schrift mit Notepad

Ist es Ihnen nicht manchmal auch so ergangen, daß Sie in irgendeiner Form mal eine nettere und größere Schrift benötigten, ohne vorher umständliche Schriftenprogramme zu laden oder sich durch Malprogramme zu quälen? Das Notepad macht es, besonders da jetzt ja der "Ausbügel-effekt" besteht. Schön und gut. Ich jedenfalls benötigte nur mal so ganz einfach eine größere Menge von groß beschrifteten Aktenordner-Rücken. Der Effekt unter dem 'Projekt=Print Large' war nicht so übel, doch noch immer zu klein. Selbst mit dem größtmöglichen Font (Diamond 20) hat es mich nicht vom Hocker gerissen. Und dann kam mir eine Idee. Ich verkleinerte das Notepad-Fenster, so daß gerade noch die gewünschte Schrift zum Vorschein kam und dabei passierte es, daß die Schrift so vergrößert wurde, daß ich zufrieden war. Wem kann dies helfen? Ich spreche hier besonders die Schüler an, die sich nicht gleich ein teures Malprogramm kaufen können, hier haben Sie mit einfacher und simpler Methode einen kleinen Zeichenclou.

P.S.: Wer jetzt übertreiben will, legt noch eine Font-Diskette ein, die eine extra große Schrift (Helvetica 40) bietet. Doch dann können Sie gleich Verhandlungen mit einer Firma führen, die Litfaßsäulen-Papier liefert!

(Gerd Hunger/jb)

## Besseres INPUT in GFA-Basic

Der am meisten gebrauchte Befehl, um Eingaben vom Benutzer zu holen, ist sowohl in

Amiga-BASIC als auch in anderen BASIC-Dialekten der mehr oder weniger komfortable Befehl INPUT. Der INPUT-Befehl in GFA-BASIC hat seinem Amiga-BASIC-Kollegen einiges voraus:

a) In GFA-BASIC kann man den zur Eingabe auffordernden Text nicht mehr einfach mit der Backspace-Taste löschen.

b) In GFA-BASIC kann man mit den Cursor-Tasten im Text herumspringen und so sehr viel einfacher Korrekturen vornehmen.

Allerdings hat auch der GFA-BASIC-INPUT-Befehl zwei Schwächen:

a) Nach dem zur Eingabe auffordernden Text erscheint immer das lästige Fragezeichen, zum Beispiel: INPUT "Eingabe: ";ein\$ ergibt auf dem Bildschirm 'Eingabe:?'.

b) Wenn man in einer durch INPUT erzeugten Abfrage Text eingibt und die BACKSPACE-Taste betätigt, wird rechts von der Eingabe befindliche Grafik mit dem Cursor nach links gezogen:

Grafiklinie

Eingabe: ? Testeingab

Der Benutzer hat sich vertippt und betätigt die BACKSPACE-Taste:

Eingabe: ? Testeing

Selbst wenn jetzt der berichtete Text weiter eingegeben wird, wird die Grafik nicht wieder nach rechts eingerückt.

### ABHILFE:

a) Um das lästige Fragezeichen loszuwerden, muß nur die Syntax des INPUT-Befehls etwas geändert werden:

INPUT "Eingabe: ",ein\$

Wird also statt des Semikolons ein Komma benutzt, wird das Fragezeichen unterbunden.

b) Die Lösung des zweiten Problems ist etwas komplizierter. Hier kann man leicht auf ein Hilfsprogramm zurückgreifen, wenn der Problemfall vorliegt. Wie gesagt, es ist nur ein Hilfsprogramm, bei dem, um die Übersichtlichkeit so groß wie möglich zu halten, auf die Bedienung durch die Cursor-Tasten verzichtet werden mußte. Dafür verfügt SMART\_INPUT, so heißt unser Hilfsprogramm, über andere Optionen, die die

### Wichtiger Hinweis:

Da manche Listings, Dateien und Listen zu sehr in die Breite gehen, sind wir manchmal gezwungen, Zeileneinrückungen fallenzulassen, um mit der Breite einer Textspalte nicht in Konflikt zu kommen. Damit Sie beim Eingeben keinerlei Probleme haben, werden alle Dateien mit Zeilennummern versehen. Diese dienen nur der besseren Übersicht und sollten nicht mit übernommen werden. Bei der Eingabe der Dateien halten Sie sich bitte an die entsprechenden Vorgaben, für die jeweiligen Programme. Mountlists sollten so eingegeben werden, wie sie im DEVS-Verzeichnis stehen.

(Red.)

Eingabe eines Textes komfortabler gestalten:

Das mit MERGE einzubindende Unterprogramm 'SMART\_INPUT' verlangt sechs Parameter. Der Aufruf sieht so aus:

```
SMART_INPUT(x-Position
%,y-Position%,Text$,MIN%,
MAX%,geheim%)
```

Zu den Parametern:

x-Position: die x-Koordinate, an der der Text erscheint, der zur Eingabe auffordert (Integer)

y-Position: die y-Koordinate, eine Einheit wie Spalte, Zeile (Integer)

Text: Text, der zur Eingabe auffordert, zum Beispiel "Ihr Name:" (String)

MIN: vom Programmierer festgelegte Mindestlänge des einzugebenden Textes (Integer)

MAX: vom Programmierer festgelegte Höchstlänge des einzugebenden Textes (Integer)

geheim: eine Option, die es erlaubt, Text verdeckt einzugeben - bei der Eingabe erscheint statt des getippten Zeichens nur ein X (beliebig änderbar); um Text verdeckt einzugeben, muß die Variable 'geheim%' im Parameter auf TRUE (-1) gesetzt werden (Integer).

Das Unterprogramm SMART\_INPUT kann den individuellen Bedürfnissen des Benutzers angepaßt werden, indem man zum Beispiel das Cursor-Zeichen oder die ausgegebenen Zeichen im Geheim-Modus ändert. Ein weiterer GFA-BASIC-Befehl zur Texteingabe ist der Befehl INPUT\$ mit der Syntax:

```
ein$ = INPUT$(anzahl)
```

Dieser Befehl liest 'anzahl' Zeichen von der Tastatur und legt sie im String 'ein\$' ab. Während der gesamten Eingabe wird kein Zeichen auf dem Bildschirm ausgegeben. Dieses Beispiel der verdeckten Texteingabe hat aber auch einen Nachteil, der durch SMART\_INPUT gelöst werden kann: Wenn die durch 'anzahl' bestimmte Höchstmenge einzugebender Zeichen erreicht ist, behandelt GFA-BASIC die Eingabe als abgeschlossen und beendet die Eingabe, ohne dem Benutzer die Möglichkeit zu geben, seinen Text zu verbessern (blind). SMART\_INPUT schafft auch hier Abhilfe: Wenn man im Programm das Zeichen 'X' durch ein Leerzeichen <SPACE> ersetzt und im Parameter den Geheim-Modus aktiviert (durch TRUE), so wird der

String ebenfalls verdeckt eingegeben, ohne daß man sehen könnte, wieviele Zeichen getippt wurden. Allerdings bleibt dem Benutzer auch nach Erreichen der maximalen Zeichenanzahl immer noch die Möglichkeit, seine Eingabe zu verbessern. Außerdem gibt es vor allem einem Anfänger das Gefühl, das Programm besser unter Kontrolle zu haben, als wenn der Computer die Eingabe selbständig beendet.

(Boris Funke/jb)

## Schneller mit der RAM-Disk

Wenn man mit dem CLI arbeitet, muß man immer etwas länger warten, da die entsprechenden Befehle erst von der Workbench geladen werden müssen. Wenn man nun aber die Befehle in die RAM-Disk kopiert, kann man mit dem CLI um einiges schneller arbeiten. Man kopiert das Verzeichnis C der Workbench so in die RAM-Disk (für Amiga mit 512 kByte): `makedir ram:c`

(Verzeichnis C im RAM erstellen)

```
copy c/copy to ram:c
copy c/dir to ram:c
copy c/cd to ram:c
copy c/info to ram:c
copy c/list to ram:c
```

...und noch weitere Befehle, die man hin und wieder braucht. Hat man 1 MByte Speicher zur Verfügung, kann man das gesamte C-Verzeichnis ins RAM kopieren:

```
copy c to ram:c
```

(Sascha Endlein/jb)

## Guter Rat fürs RAD:!

Für die optimale Nutzung ihrer RAD: sollten sie das FAST-FILING-SYSTEM (Beschleunigung des Zugriffs) nutzen. Hierzu sind lediglich einige Änderungen in der Mountlist (Verzeichnis "devs") notwendig. Achten Sie bitte darauf, daß Ihre Bootpriorität auf -128 festgelegt ist und Surfaces, BlockPerTrack, LowCyl und HighCyl den Werten eines Diskettenlaufwerks entsprechen. Die RAD: in der Mountlist sieht nun wie folgt aus:

```
1 RAD: Device = ramdrive
      .device
2 Unit = 1
3 Flags = 0
4 Surfaces = 2
5 BlockPerTrack = 11
6 Reserved = 2
7 Interleave = 0
8 LowCyl = 0 ; HighCyl = 79
9 Buffers = 5
10 BufMemType = 1
11 GlobVec = -1
12 StackSize = 4000
13 DosType = 0x444F5301
14 FileSystem = L:fastfilesy
      stem
15 BootPri = -128
16 #
```

(Die Zahlen vor den Zeilen dürfen natürlich nicht mit übernommen werden, sie dienen nur der Übersichtlichkeit, Red.)

Aufgrund der Bootpriorität von -128 ist nun kein Autoboot mehr möglich.

## REM-Ersatz

In Amiga-BASIC wird wie in anderen BASIC-Dialekten auch für Anmerkungen der Befehl REM (REMark) verwendet. Man kann den gleichen Befehl aber auch durch Setzen eines Apostrophs erreichen. Der Editor erkennt allerdings nur das Apostroph, das über die Tastenkombination <ALT> + <ä> aufgerufen wird, als ein REM an; die auf der Tastatur vermerkten funktionieren hierbei nicht.

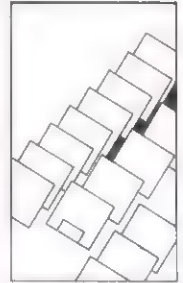
(Christian Czerwinski)

## Icons für PD-Disketten

Bei älteren Public-Domain-Disketten kann ab und zu das Problem auftauchen, daß die enthaltenen Programme sich nicht von der Workbench laden lassen, da die entsprechenden Programme keine Icons besitzen und man sie somit nicht komfortabel starten kann. Mein kleines Programm sorgt hier für Abhilfe, obwohl es nur sieben Zeilen lang ist. Geschrieben wurde es in GFA-BASIC und sollte möglichst mit dem GFA-BASIC-Compiler kompiliert werden, damit es sinnvoll eingesetzt werden kann. Danach ist es in die Startup-Sequence einzubinden, damit später die entsprechenden Programme mit der Maus gestartet werden können. Hier nun das Programm:

```
1 start:
2 fileselect "SYS:", "EXECUTE"
  ,"" ,la$
3 if la$ = "" then
4 end
5 endif
6 exec la$,-1,-1
7 goto start
```

(Manfred Dittrich jun./jb)



## Tausche Tip gegen DATABOX

Haben Ihnen die Kurz-Tips auf unseren Seiten gefallen? Vielleicht haben Sie selbst einen Tip parat, der andere Amiga-Besitzer interessieren könnte? Dann machen wir Ihnen wieder unser Angebot: Sie schicken einen oder mehrere kurze Tips an uns, und wenn diese(r) uns gefällt(en), bekommen Sie von uns im Austausch die DATABOX desjenigen Heftes zugeschickt, in dem Ihr Tip erscheint.

Dabei sollten Sie noch folgendes beachten:

- Schreiben Sie Ihren Tip als ASCII-Datei auf Ihre Diskette, möglichst ohne eigensinnige Verschnörkelungen, da alle Texte zur weiteren Verarbeitung auf Personal-Computer transferiert werden müssen. Sonder- und Steuerzeichen haben dabei meist seltsame Auswirkungen.

- Schreiben Sie auf den Brief sowie auf einen kleinen Aufkleber auf der Diskette Ihre lückenlose Anschrift und einen kleinen Hinweis, was auf der Diskette zu finden ist.

Dann können Sie sicher sein, daß die DATABOX Sie erreicht. Schicken Sie das Ganze an den

DMV-Verlag  
Redaktion AMIGA DOS  
Kennwort "Tausche Tip  
gegen Databox"  
Postfach 250  
D-3440 Eschwege



## Listings

```

01 PROCEDURE smart_input(x%,y%,text$,min%,max%,geh%)
02 ' x-Koordinate, y-Koordinate, Requester-Text, minimale Ei
ngabelänge
03 ' maximale Eingabelänge, GEHEIM-Modus
04 LOCAL a%,k% : lokale Variablen
05 PRINT AT(x%,y%);text$;" _"; : Cursor auf Anfang setze
n
06 x%=x%+LEN(text$)+1 : Eingabestring initialis
ieren
07 ein$=SPACE$(max%) : Tastaturpuffer leeren
08 WHILE INKEY$<>""
09 WEND : S C H L E I F E
10 DO : Zeichen von Tastatur ho
len
11 i$=INKEY$ : ASCII-Wert bilden
12 a%=ASC(i$) : Backspace eingeben UN
Eingabestring
13 IF a%=8 AND k%>0 : mindestens ein Zeichen
lang
14 DEC x%
15 MID$(ein$,k%,1)=" " : Cursor nach links verse
tzen und String-
16 DEC k% : Zeiger verschieben
17 PRINT AT(x%,y%);" _ ";
18 GOTO weiter2
19 ELSE IF a%=196 OR a%=214 OR a%=220 OR a%=228 OR a%=246
OR a%=252
20 GOTO weiter1 : weiter, wenn Umlaut ein
gegeben
21 ELSE IF a%<32 OR a%>125 : sonst - kein gültiges A
SCII-Zeichen
22 GOTO weiter2
23 ENDIF
24 weiter1: : wenn Anzahl der erlaubt
en Zeichen noch
25 IF k%<max% : nicht erreicht, dann w
eiter
26 IF geh%=TRUE : wenn verdeckt eingegebe
n werden soll,
27 PRINT AT(x%,y%);"X"; : nur 'X' ausgeben
28 ELSE
29 PRINT AT(x%,y%);i$; : sonst eingegebenes Zeic
hen zeigen
30 ENDIF
31 IF k%<max% : wenn noch nicht vorletz
te Stelle, Cursor
32 PRINT " " : nach rechts versetzen
33 ENDIF
34 INC k% : x-Koordinate des Cursor
s und
35 INC k% : Anzahl der Zeichen um 1
erhöhen
36 MID$(ein$,k%,1)=i$ : Eingabestring == eingeg
ebenes Zeichen
37 weiter2: : erweitern
38 ENDIF
39 EXIT IF i$=CHR$(13) AND k%>min% : Schleife verlassen,
wenn RETURN ge-
40 LOOP : drückt wurde UND mindes
tens min% Zeichen
41 : eingegeben wurden
42 PRINT AT(x%,y%);" " : letzten Cursor löschen
(wichtig, wenn
43 : weniger als max% Zeiche
n eingegeben
44 : wurden, da sonst ein Cu
rsor-Zeichen
45 : stehen bliebe
46 ein$=TRIM$(ein$) : eventuelle Leerzeichen
vom Eingabestring
47 RETURN : trennen

```

Listing: SMART\_INPUT von Boris Funke  
Die Zeilennummern dienen nur der besseren Uebersicht!

```

01 /*      Programm: AReq      *
02 * Compiler: Lattice Amiga C Compiler V5.02 *
03 * Autor   : Thomas Knoth   *
04 *      *                    *
05 * Copyright = 1989, 1990 by Th. Knoth, Bremen *
06 *      *                    *
07 *      */
08 #include <proto/dos.h>
09 #include <dos.h>
10 #include <proto/exec.h>
11 #include <intuition/intuition.h>
12
13 struct IntuitionBase *IntuitionBase;
14
15 BOOL req(char *,char *, char *);

```

Listing: die 'Gewußt viel'-Programme

```

16
17 struct IntuiText Kasten[3] =
18 {
19     0, 1, JAM1,10,10, NULL, NULL, NULL,
20     2, 3, JAM1, 5, 3, NULL, NULL, NULL,
21     2, 3, JAM1, 5, 3, NULL, NULL, NULL
22 };
23
24 void main(argc, argv)
25 SHORT argc;
26 signed char *argv[];
27 {
28     if(argc==4)
29     {
30         IntuitionBase=(struct IntuitionBase*)OpenLibrary("int
uition.library", 0L);
31
32         if((req(argv[1],argv[2], argv[3]))==TRUE)
33         {
34             CloseLibrary(IntuitionBase);
35             exit(RETURN_WARN);
36         }
37         else
38         {
39             CloseLibrary(IntuitionBase);
40             exit(RETURN_OK);
41         }
42     }
43     else
44     {
45         (void)Write(Output(),"Aufruf : AReq Body LGadget RGad
get\n",35);
46         (void)Write(Output(),"Copyright (c) 1989, 1990 by Th.
Knoth, Bremen\n",45);
47         exit(RETURN_ERROR);
48     }
49 }
50
51 BOOL req(title, yes, no)
52 char *title,*yes,*no;
53 {
54     Kasten[0].IText=(UBYTE *)title;
55     Kasten[2].IText=(UBYTE *)no;
56     Kasten[1].IText=(UBYTE *)yes;
57     return((BOOL)AutoRequest(NULL, &Kasten[0], &Kasten[1],
&Kasten[2], 0L, 0L, 320L, 60L));
58 }

```

Listing: AReq.C von Thomas Knoth

```

01 ; Beispiel Batch-Datei zum Befehl "areq"
02 ; Startet areq und fragt danach den Fehlercode ab.
03 ; areq gibt warn zurück, wenn das linke Gadget gedrueckt
wurde, in diesem
04 ; Fall "Natuerlich". Ansonsten wird der Fehlercode "0" (a
lles ok) zurueck-
05 ; gegeben.
06 ; areq kehrt mit dem Fehlercode 10 zurueck, wenn ein fals
cher Aufruf er-
07 ; folgte !
08 ;
09 ; Copyright (c) 1989, 1990 by Th. Knoth, Bremen
10
11 areq "Benutzen Sie die WB 1.3 ?" Natuerlich "NOCH nicht !"
12
13 if warn
14     echo "Er ist UpToDate !"
15 else
16     echo "Last but not least ein Oldie !"
17 endif

```

Listing: AReq- Test-Batch-File

```

01 .key datei
02
03 1c1 <datei$1>.c
04 go <datei$1>.q
05 1c2 <datei$1>.q
06 blink 11b:cres.o <datei$1>.o to <datei$1>.res 11b 11b:1
c.11b,11b:amiga.11b
07
08 ; Copyright (c) 1989, 1990 by Th. Knoth, Bremen

```

Listing: MAKE.RES-Batch-Datei

```

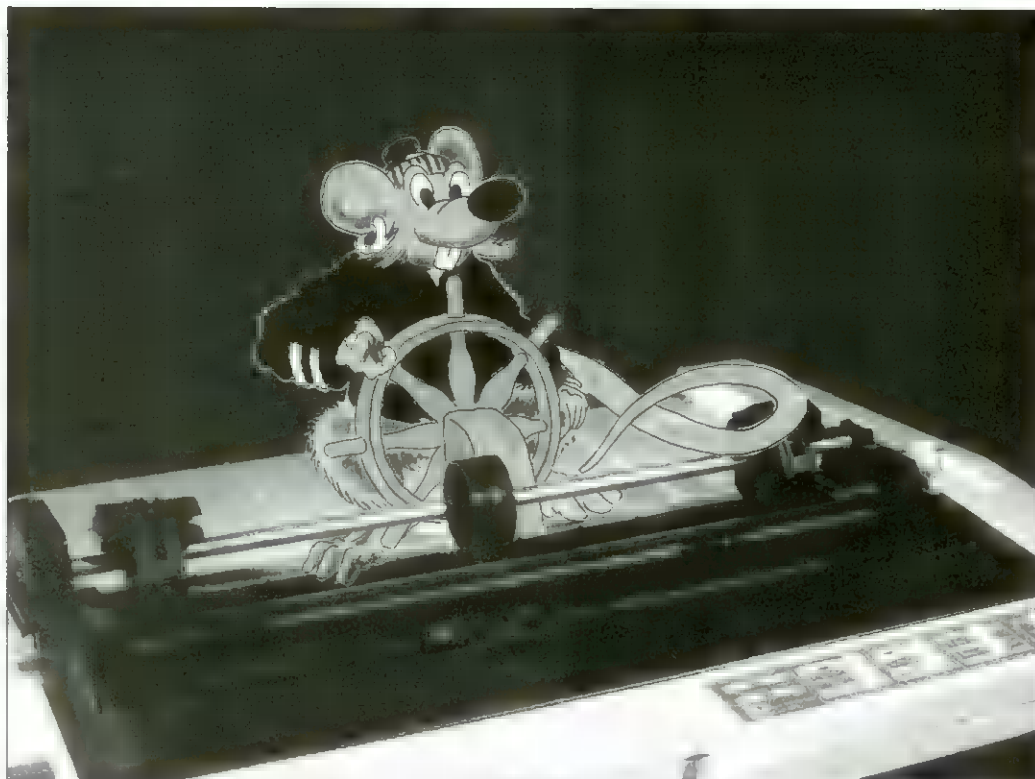
01 .key datei
02
03 1c1 <datei$1>.c
04 go <datei$1>.q
05 1c2 <datei$1>.q
06 blink 11b:c.o <datei$1>.o 11b 11b:1c.11b,11b:amiga.11b
07
08 ; Copyright (c) 1989, 1990 by Th. Knoth, Bremen

```

Listing: AReq-Make-Batch-File

Bitte die Batch- Dateien ohne Zeilennummern mit Editor ein-  
geben!

Listing: die 'Gewußt viel'-Programme



Garry Glendown

## Ruft die Mäuse auf den Plan

### Printer Control Window – Druckersteuerung per Maus

Sie kennen folgendes Problem: Oftmals hat man eben mal zwei oder drei Seiten auszudrucken, für die man ein paar Control-Codes benötigt, oder man möchte eine besondere Randeinstellung für das Ausdrucken eines Listings haben. Hat dieses File dann noch einige Standard-Epson-Controlcodes, die andere Druckereigenschaften beeinflussen, fällt die Benutzung von PRT flach. Was also tun, sprach Zeus...? Ganz einfach: nur PCW mit Run starten, die benötigten Optionen anklicken, und schon ist der Drucker einsatzbereit.

**K**ommen wir auf das Menü etwas näher zu sprechen. Ein Eintrag in der Menülste besteht aus mehreren Teilen. "Type" ist die Art des Eintrags. Im Moment existieren folgende Typen:

■ pr\_COMMAND

Die in dem Array 'Prt' abgelegten Daten sind auszudruckende Daten, die Länge steht in 'PLen'.

● pr\_EXTERN

Es wird ein externer Befehl ausgeführt, dessen Nummer in 'PLen' steht.

■ pr\_NONE

Auf Anwahl dieses Menüpunkts erfolgt keine Reaktion, er dient lediglich der Information.

"Name" ist ein String, in dem der zu druckende Text steht. Der Text darf maximal 'MAXLINE'-Zeichen enthalten.

"Len" ist die Länge des zu druckenden Strings und wird vom Programm eingefügt (in der Routine 'OpenStuff()', siehe Remarks).

"Prt" ist ein Array, das die zu druckenden Zeichen enthält. "PLen" gibt die Anzahl der zu druckenden Zeichen bzw. die Nummer der Extern-Routine wieder.

"FPen" und "BPen" stellen die Vorder- und Hintergrundfarbe des Textes im Menü dar. Die Anzahl der Menüpunkte muß ein Vielfaches von sechs sein, da jeweils sechs Menüpunkte auf einmal angezeigt

werden. Man kann zwar das Menüprogramm so umändern, daß es beim Scrollen um jeweils eine Zeile nach oben bzw. unten geschoben wird, aber dies schien mir, vor allem in Anbetracht der teilweise doch recht großen Datenmenge, uninteressant, da das Blättern nach einem bestimmten Menüpunkt zu lang dauert.

### Nicht nur Maus – nein, auch Taste

Alle Menüpunkte können auch über die Tastatur ausgewählt werden. Dabei gilt folgende Belegung:

- Tasten 1 bis 6: Auswahl der einzelnen Menüpunkte
- Crsr Dn: Blättern nach unten
- Crsr Up: Blättern nach oben
- Help: Info
- Q: Programm beenden (Requester beachten!)

Sollten bei der Initialisierung Fehler auftreten, so wird der Benutzer mittels eines Requesters darauf hingewiesen. Außer dem Requester, der bei Anwählen von 'Q' bzw. der Exit-Funktion erscheint, ist es egal, mit welchem Gadget man den Requester quittiert. Sollte man die Exit-Funktion (bzw. 'Q' auf der Tastatur) aus Versehen angewählt haben, so kann man im Programm bleiben, indem man 'Nein' anwählt (Tastatur: LeftAmiga-V), durch Selektieren von 'OK' (LeftAmiga-B) wird PCW verlassen.

### Auf, ans Eingemachte

Deaktiviert man das PCW-Window (durch Anklicken eines anderen Fensters), wird es in Y-Richtung auf wenige Pixel verkleinert, um so wenig Platz wie möglich auf dem Bildschirm zu belegen. Man kann es natürlich auch mittels Depth("Tiefen")-Gadget hinter andere Windows legen, doch ist es recht praktisch, wenn es jederzeit ohne große Sucherei anklickbar ist. Durch die Verkleinerung nimmt es in Y-Richtung genausoviel Platz ein wie die Menüleiste eines Windows, so daß man es bequem in die linke oder rechte obere Ecke legen kann.

Im Zusammenhang mit dem Vergrößern des Windows auf Normalgröße fällt eine Eigenheit von Intuition auf, die ich schon als 'Bug' bezeichne: Es kommt nämlich die Intui-Meldung 'ACTIVEWINDOW', wenn das Window verschoben wird und vorher inaktiv war. Diese Meldung kommt aber leider zu früh, sofort nach Anklicken des Windows, obwohl das Window noch bewegt wird. Greift man dann auf Teile der Window-Struktur zu, so enthält sie die (für diese Verwendung) falschen Werte.

Das führt dann dazu, daß das Window aus dem Bildschirm heraus'gesized' wird, was meist zu einem nicht so schönen



nen Bildaufbau führt (oder wie es im Addison-Wesley-Buch heißt: "Wegen Verzerrungen im Raum-Zeit-Kontinuum, die daraus resultieren können, wie es in der speziellen Relativitätstheorie beschrieben ist, bietet das Ergebnis gewöhnlich keinen schönen Anblick"). Daher ist im Programm darauf zu warten, daß der linke Mausknopf losgelassen wird, um erst danach (wenn das Window an seiner endgültigen Position liegt) auf die Window-Koordinaten zuzugreifen. Zusätzlich dazu wartet man im Verlauf der Size-Routine noch mehrmals auf Intui-Ticks, also jeweils n/10 Sekunden, um sicherzugehen, daß die Operationen wie MoveWindow und SizeWindow fertig durchgeführt sind.

## Jetzt passiert's ...

Wird ein Menüpunkt ausgewählt, ruft das Hauptprogramm die Routine 'Dolt' zu-

sammen mit der Nummer der Auswahl auf. Diese Routine wickelt dann alle anderen Funktionen ab. Somit beschränkt sich das Hauptprogramm lediglich auf das Abholen und Auswerten der IntuiMessages.

"Dolt" verzweigt als erstes nach dem Typ der Auswahl, also "pr\_Command", "pr\_Extern" oder "pr\_None" (letzteres wird eigentlich einfach übergangen). Handelt es sich bei der Auswahl um einen Druckerbefehl, so wird der Druckerkanal geöffnet, der String ausgegeben, ein "BEEP" (CHR\$(7)) als Bestätigung des Kommandos angehängt und der Druckerkanal wieder geschlossen. Im Falle von "pr\_Extern" wird nach der Nummer von "men[n].Plen" verzweigt. In dem abgedruckten Programm steht dabei '1' für Quit, '2' für Info und '3' für Rückkehr ins normale Menü.

Hier kann das Programm natürlich beliebig erweitert wer-

den, zum Beispiel durch Einfügung des Ausdrucks Files oder anderen Utilities. Die einzige Beschränkung besteht darin, daß man nur 256 verschiedene Extern-Routinen definieren kann, da "Plen" als BYTE definiert ist (Klar, man kann das in 'Int' oder gar 'long' ändern, aber ich habe noch kein Programm mit über vier Milliarden Optionen gesehen; abgesehen davon bedeutet das Optionen von 6 Seiten. Viel Vergnügen beim Durchblättern...). Auch ohne Änderung von "Plen" kann man noch mehr Routinen einbauen, nämlich durch Einführen eines neuen Typs, zum Beispiel "pr\_Extern2" (Wer mehr als 512 Funktionen benötigt, sollte mal seinen Programmierstil überprüfen oder sich eine(n[?]) Cray 2 anschaffen...).

## Hosen runter...

Wie im Skat muß auch hier das Blatt aufgedeckt werden.

allerdings handelt es sich hier um die Menüpunkte. Dies geschieht in der "Display\_em" Routine. Sie übernimmt alles von der Positionierung des Cursors über die Farbeinstellung bis hin zum Ausdrucken der Texte. Vorher wird der Bildschirmbereich gelöscht.

Zum Schluß noch ein Hinweis: Alle Drucker-Codes im abgedruckten Programm gelten für Drucker der NEC-Pinwriter-Serie sowie Kompatible (Epson Q-Serie etc.). Das heißt, falls Sie Ihren Drucker mit einem Treiber wie "CBM\_MPS2xxx", "Epson\_Q" oder "NEC\_Pinwriter" betreiben, können Sie das Programm ohne Änderung übernehmen.

Andernfalls sollten Sie versuchen, die Codes entweder direkt mit dem Programm auf korrekte Funktion zu testen, oder aber die richtigen Codes mittels Blick ins Handbuch einsetzen.

(vb)

## Listings

```
1: /******
2: /*      prcw.c
3: /*      Printer Control Window V 2.0
4: /*      (C) 1990 by G.Glendown & AMIGA DOS
5: /*      Sprache: Aztec C
6: /******
7:
8: /*
9: Achtung! Die in diesem Programm verwendeten Drucker-
10: Codes sind gültig für die NEC-Pinwriter-Drucker und
11: Kompatible. Bei anderen Druckern können einzelne Codes
12: falsch oder gar nicht funktionieren. Konsultieren Sie
13: daher im Zweifelsfall ihr Handbuch, um zu sehen, ob
14: und wie Ihr Drucker die abgedruckten Funktionen ver-
15: wendet.
16: Compiler-Hinweis: Alle Includes sind vorcompiliert,
17: das heißt, in einem File abgelegt, so daß lediglich
18: ein File eingelesen werden muß. Um dieses File zu er-
19: zeugen, muß man ein Source-File schreiben, das alle
20: Include-Files mit #include einliest und dieses mit:
21: cc (name).c -a +include-name)
22: compiliert. Das fertige File wird dann beim Compi-
23: lieren mit:
24: cc (prg).c +include-name)
25: eingebunden. Zum Linken wird lediglich die c.lib-
26: library benötigt.
27:
28: #define USB (UBYTE *)
29: #define pr_COMMAND 0x01
30: #define pr_EXTERN 0x02
31: #define pr_NONE 0x00
32: #define ESC 27
33: #define FS 28
34: #define MAXLINE 21
35:
36: #include "pcw_defs.h"
37:
38: struct prt {
39:     BYTE Type; /* Art des Eintrags:
40:                pr_Command/Extern/None */
41:     UBYTE Name[MAXLINE]; /* Auszudruckender Name */
42:     BYTE Len; /* Länge des Texts */
43:     UBYTE Prt[10]; /* Prt-Codes */
44:     BYTE Plen; /* Länge Prt-Codes
45:                bzw. Extern-Routine */
46:     UBYTE FPen, BPen; /* Vorder-/Hintergrundfarbe */
47: };
48:
49: struct prt info[] = {
50:     /* Info, wird als Menü gehandhabt.
51:     Sobald ein Menüpunkt ausgewählt
52:     wird, wird in den normalen Teil
53:     zurückgeschaltet.
54:     */
55:     /* Extern Routine #3:
56:     Rückkehr ins normale Menü
57:     */
58:     pr_EXTERN, "Dieses Programm ist", 0, 3, 1, 0,
59:     pr_EXTERN, "komplett mausge-", 0, 3, 1, 0,
60:     pr_EXTERN, "steuert. Zum Aus-", 0, 3, 1, 0,
61:     pr_EXTERN, "wählen der Seite", 0, 3, 1, 0,
62:     pr_EXTERN, "einfach einen der", 0, 3, 1, 0,
63:     pr_EXTERN, "Pfeile anklicken", 0, 3, 1, 0,
64:     pr_EXTERN, "Zum Auswählen einer", 0, 3, 1, 0,
65:     pr_EXTERN, "Option diese nur", 0, 3, 1, 0,
66:     pr_EXTERN, "mit der Maus an-", 0, 3, 1, 0,
67:     pr_EXTERN, "klicken. Rückkehr", 0, 3, 1, 0,
68:     pr_EXTERN, "zum Hauptmenü durch", 0, 3, 1, 0,
69:     pr_EXTERN, "irgendeine Option", 0, 3, 1, 0,
70:     pr_EXTERN, "Alle Optionen", 0, 3, 1, 0,
71:     pr_EXTERN, "können auch mit der", 0, 3, 1, 0,
72:     pr_EXTERN, "Tastatur gewählt", 0, 3, 1, 0,
73:     pr_EXTERN, "werden", 0, 3, 1, 0,
74:     pr_EXTERN, "Taste Funktion:", 0, 3, 1, 0,
75:     pr_EXTERN, "1-6 Menüpunkt", 0, 3, 1, 0,
76:     pr_EXTERN, "Crshr up Seite", 0, 3, 1, 0,
77:     pr_EXTERN, "Crshr dn Seite", 0, 3, 1, 0,
78:     pr_EXTERN, "Q Ende", 0, 3, 1, 0,
79:     pr_EXTERN, "0", 0, 3, 1, 0,
80:     #define NUM_INFO
81:     struct prt menu[] = {
82:         pr_COMMAND, "Italic on", 0, {ESC, 'I'}, 2, 1, 0,
83:         pr_COMMAND, "Italic off", 0, {ESC, 'I'}, 2, 1, 0,
84:         pr_COMMAND, "Double strike on", 0, {ESC, 'G'}, 2, 1, 0,
85:         pr_COMMAND, "Double strike off", 0, {ESC, 'G'}, 2, 1, 0,
86:         pr_COMMAND, "Underline On", 0, {ESC, 'U'}, 3, 1, 0,
87:         pr_COMMAND, "Underline Off", 0, {ESC, 'U'}, 3, 1, 0,
88:         pr_COMMAND, "10 CPI", 0, {18, ESC, 'P'}, 3, 1, 0,
89:         pr_COMMAND, "12 CPI", 0, {18, FS, 'S'}, 6, 1, 0,
90:         pr_COMMAND, "High Speed", 0, {18, ESC, 'M', FS, 'S'}, 6, 1, 0,
91:         pr_COMMAND, "15 CPI", 0, {18, ESC, 'P'}, 3, 1, 0,
92:         pr_COMMAND, "17 CPI", 0, {ESC, 'P', 15}, 3, 1, 0,
93:         pr_COMMAND, "20 CPI", 0, {ESC, 'P', 15}, 3, 1, 0,
94:         pr_COMMAND, "CS: USA", 0, {ESC, 'R', 0}, 3, 1, 0,
95:         pr_COMMAND, "CS: D", 0, {ESC, 'R', 2}, 3, 1, 0,
96:         pr_COMMAND, "CS: IBM", 0, {FS, 'I', 1}, 3, 1, 0,
97:         pr_COMMAND, "6 LPI", 0, {ESC, 'L', 2}, 2, 1, 0,
98:         pr_COMMAND, "8 LPI", 0, {ESC, 'L', 2}, 2, 1, 0,
99:         pr_COMMAND, "LF: 1/18", 0, {ESC, 'L', 10}, 3, 1, 0,
100:         pr_COMMAND, "Superscript", 0, {ESC, 'S', 0}, 3, 1, 0,
101:         pr_COMMAND, "Subscript", 0, {ESC, 'S', 1}, 3, 1, 0,
102:         pr_COMMAND, "Cancel Sub/Superscr.", 0, {ESC, 'T'}, 2, 1, 0,
103:         pr_COMMAND, "FF", 0, {12}, 1, 1, 0,
104:         pr_COMMAND, "SO on", 0, {ESC, 'W', 1}, 3, 1, 0,
105:         pr_COMMAND, "SO off", 0, {ESC, 'W', 0}, 3, 1, 0,
106:         pr_COMMAND, "Left Justify", 0, {ESC, 'J', 0}, 3, 1, 0,
107:         pr_COMMAND, "Right Justify", 0, {ESC, 'J', 1}, 3, 1, 0,
108:         pr_COMMAND, "Center Justify", 0, {ESC, 'J', 2}, 3, 1, 0,
109:         pr_COMMAND, "Full Justify", 0, {ESC, 'J', 3}, 3, 1, 0,
110:         pr_COMMAND, "Proportional on", 0, {ESC, 'P', 1}, 3, 1, 0,
111:         pr_COMMAND, "Proportional off", 0, {ESC, 'P', 0}, 3, 1, 0,
112:         pr_COMMAND, "Draft", 0, {ESC, 'X', 0}, 3, 1, 0,
113:         pr_COMMAND, "LQ", 0, {ESC, 'X', 1}, 3, 1, 0,
114:         pr_COMMAND, "Half Speed", 0, {ESC, 'S', 1}, 3, 1, 0,
115:         pr_COMMAND, "Full Speed", 0, {ESC, 'S', 0}, 3, 1, 0,
116:         pr_COMMAND, "User-defined CS on", 0, {ESC, 'W', 1}, 3, 1, 0,
117:         pr_COMMAND, "User-defined CS off", 0, {ESC, 'W', 0}, 3, 1, 0,
118:     };
119:     };
120:     pr_COMMAND, "User-defined CS on", 0, {ESC, 'W', 1}, 3, 1, 0,
121:     pr_COMMAND, "User-defined CS off", 0, {ESC, 'W', 0}, 3, 1, 0,
122:     };
123:     };
124:     };
125:     };
126:     };
127:     };
128:     };
129:     };
130:     };
131:     };
132:     };
133:     };
134:     };
135:     };
136:     };
137:     };
138:     };
139:     };
140:     };
141:     };
142:     };
143:     };
144:     };
145:     };
146:     };
147:     };
148:     };
149:     };
150:     };
151:     };
152:     };
153:     };
154:     };
155:     };
156:     };
157:     };
158:     };
159:     };
160:     };
161:     };
162:     };
163:     };
164:     };
165:     };
166:     };
167:     };
168:     };
169:     };
170:     };
171:     };
172:     };
173:     };
174:     };
175:     };
176:     };
177:     };
178:     };
179:     };
180:     };
181:     };
182:     };
183:     };
184:     };
185:     };
186:     };
187:     };
188:     };
189:     };
190:     };
191:     };
192:     };
193:     };
194:     };
195:     };
196:     };
197:     };
198:     };
199:     };
200:     };
201:     };
202:     };
203:     };
204:     };
205:     };
206:     };
207:     };
208:     };
209:     };
210:     };
211:     };
212:     };
213:     };
214:     };
215:     };
216:     };
217:     };
218:     };
219:     };
220:     };
221:     };
222:     };
223:     };
224:     };
225:     };
226:     };
227:     };
228:     };
229:     };
230:     };
231:     };
232:     };
233:     };
234:     };
235:     };
236:     };
237:     };
238:     };
239:     };
240:     };
241:     };
242:     };
243:     };
244:     };
245:     };
246:     };
247:     };
248:     };
249:     };
250:     };
251:     };
252:     };
253:     };
254:     };
255:     };
256:     };
257:     };
258:     };
259:     };
260:     };
261:     };
262:     };
263:     };
264:     };
265:     };
266:     };
267:     };
268:     };
269:     };
270:     };
271:     };
272:     };
273:     };
274:     };
275:     };
276:     };
277:     };
278:     };
279:     };
280:     };
281:     };
282:     };
283:     };
284:     };
285:     };
286:     };
287:     };
288:     };
289:     };
290:     };
291:     };
292:     };
293:     };
294:     };
295:     };
296:     };
297:     };
298:     };
299:     };
300:     };
301:     };
302:     };
303:     };
304:     };
305:     };
306:     };
307:     };
308:     };
309:     };
310:     };
311:     };
312:     };
313:     };
314:     };
315:     };
316:     };
317:     };
318:     };
319:     };
320:     };
321:     };
322:     };
323:     };
324:     };
325:     };
326:     };
327:     };
328:     };
329:     };
330:     };
331:     };
332:     };
333:     };
334:     };
335:     };
336:     };
337:     };
338:     };
339:     };
340:     };
341:     };
342:     };
343:     };
344:     };
345:     };
346:     };
347:     };
348:     };
349:     };
350:     };
351:     };
352:     };
353:     };
354:     };
355:     };
356:     };
357:     };
358:     };
359:     };
360:     };
361:     };
362:     };
363:     };
364:     };
365:     };
366:     };
367:     };
368:     };
369:     };
370:     };
371:     };
372:     };
373:     };
374:     };
375:     };
376:     };
377:     };
378:     };
379:     };
380:     };
381:     };
382:     };
383:     };
384:     };
385:     };
386:     };
387:     };
388:     };
389:     };
390:     };
391:     };
392:     };
393:     };
394:     };
395:     };
396:     };
397:     };
398:     };
399:     };
400:     };
401:     };
402:     };
403:     };
404:     };
405:     };
406:     };
407:     };
408:     };
409:     };
410:     };
411:     };
412:     };
413:     };
414:     };
415:     };
416:     };
417:     };
418:     };
419:     };
420:     };
421:     };
422:     };
423:     };
424:     };
425:     };
426:     };
427:     };
428:     };
429:     };
430:     };
431:     };
432:     };
433:     };
434:     };
435:     };
436:     };
437:     };
438:     };
439:     };
440:     };
441:     };
442:     };
443:     };
444:     };
445:     };
446:     };
447:     };
448:     };
449:     };
450:     };
451:     };
452:     };
453:     };
454:     };
455:     };
456:     };
457:     };
458:     };
459:     };
460:     };
461:     };
462:     };
463:     };
464:     };
465:     };
466:     };
467:     };
468:     };
469:     };
470:     };
471:     };
472:     };
473:     };
474:     };
475:     };
476:     };
477:     };
478:     };
479:     };
480:     };
481:     };
482:     };
483:     };
484:     };
485:     };
486:     };
487:     };
488:     };
489:     };
490:     };
491:     };
492:     };
493:     };
494:     };
495:     };
496:     };
497:     };
498:     };
499:     };
500:     };
501:     };
502:     };
503:     };
504:     };
505:     };
506:     };
507:     };
508:     };
509:     };
510:     };
511:     };
512:     };
513:     };
514:     };
515:     };
516:     };
517:     };
518:     };
519:     };
520:     };
521:     };
522:     };
523:     };
524:     };
525:     };
526:     };
527:     };
528:     };
529:     };
530:     };
531:     };
532:     };
533:     };
534:     };
535:     };
536:     };
537:     };
538:     };
539:     };
540:     };
541:     };
542:     };
543:     };
544:     };
545:     };
546:     };
547:     };
548:     };
549:     };
550:     };
551:     };
552:     };
553:     };
554:     };
555:     };
556:     };
557:     };
558:     };
559:     };
560:     };
561:     };
562:     };
563:     };
564:     };
565:     };
566:     };
567:     };
568:     };
569:     };
570:     };
571:     };
572:     };
573:     };
574:     };
575:     };
576:     };
577:     };
578:     };
579:     };
580:     };
581:     };
582:     };
583:     };
584:     };
585:     };
586:     };
587:     };
588:     };
589:     };
590:     };
591:     };
592:     };
593:     };
594:     };
595:     };
596:     };
597:     };
598:     };
599:     };
600:     };
601:     };
602:     };
603:     };
604:     };
605:     };
606:     };
607:     };
608:     };
609:     };
610:     };
611:     };
612:     };
613:     };
614:     };
615:     };
616:     };
617:     };
618:     };
619:     };
620:     };
621:     };
622:     };
623:     };
624:     };
625:     };
626:     };
627:     };
628:     };
629:     };
630:     };
631:     };
632:     };
633:     };
634:     };
635:     };
636:     };
637:     };
638:     };
639:     };
640:     };
641:     };
642:     };
643:     };
644:     };
645:     };
646:     };
647:     };
648:     };
649:     };
650:     };
651:     };
652:     };
653:     };
654:     };
655:     };
656:     };
657:     };
658:     };
659:     };
660:     };
661:     };
662:     };
663:     };
664:     };
665:     };
666:     };
667:     };
668:     };
669:     };
670:     };
671:     };
672:     };
673:     };
674:     };
675:     };
676:     };
677:     };
678:     };
679:     };
680:     };
681:     };
682:     };
683:     };
684:     };
685:     };
686:     };
687:     };
688:     };
689:     };
690:     };
691:     };
692:     };
693:     };
694:     };
695:     };
696:     };
697:     };
698:     };
699:     };
700:     };
701:     };
702:     };
703:     };
704:     };
705:     };
706:     };
707:     };
708:     };
709:     };
710:     };
711:     };
712:     };
713:     };
714:     };
715:     };
716:     };
717:     };
718:     };
719:     };
720:     };
721:     };
722:     };
723:     };
724:     };
725:     };
726:     };
727:     };
728:     };
729:     };
730:     };
731:     };
732:     };
733:     };
734:     };
735:     };
736:     };
737:     };
738:     };
739:     };
740:     };
741:     };
742:     };
743:     };
744:     };
745:     };
746:     };
747:     };
748:     };
749:     };
750:     };
751:     };
752:     };
753:     };
754:     };
755:     };
756:     };
757:     };
758:     };
759:     };
760:     };
761:     };
762:     };
763:     };
764:     };
765:     };
766:     };
767:     };
768:     };
769:     };
770:     };
771:     };
772:     };
773:     };
774:     };
775:     };
776:     };
777:     };
778:     };
779:     };
780:     };
781:     };
782:     };
783:     };
784:     };
785:     };
786:     };
787:     };
788:     };
789:     };
790:     };
791:     };
792:     };
793:     };
794:     };
795:     };
796:     };
797:     };
798:     };
799:     };
800:     };
801:     };
802:     };
803:     };
804:     };
805:     };
806:     };
807:     };
808:     };
809:     };
810:     };
811:     };
812:     };
813:     };
814:     };
815:     };
816:     };
817:     };
818:     };
819:     };
820:     };
821:     };
822:     };
823:     };
824:     };
825:     };
826:     };
827:     };
828:     };
829:     };
830:     };
831:     };
832:     };
833:     };
834:     };
835:     };
836:     };
837:     };
838:     };
839:     };
840:     };
841:     };
842:     };
843:     };
844:     };
845:     };
846:     };
847:     };
848:     };
849:     };
850:     };
851:     };
852:     };
853:     };
854:     };
855:     };
856:     };
857:     };
858:     };
859:     };
860:     };
861:     };
862:     };
863:     };
864:     };
865:     };
866:     };
867:     };
868:     };
869:     };
870:     };
871:     };
872:     };
873:     };
874:     };
875:     };
876:     };
877:     };
878:     };
879:     };
880:     };
881:     };
882:     };
883:     };
884:     };
885:     };
886:     };
887:     };
888:     };
889:     };
890:     };
891:     };
892:     };
893:     };
894:     };
895:     };
896:     };
897:     };
898:     };
899:     };
900:     };
901:     };
902:     };
903:     };
904:     };
905:     };
906:     };
907:     };
908:     };
909:     };
910:     };
911:     };
912:     };
913:     };
914:     };
915:     };
916:     };
917:     };
918:     };
919:     };
920:     };
921:     };
922:     };
923:     };
924:     };
925:     };
926:     };
927:     };
928:     };
929:     };
930:     };
931:     };
932:     };
933:     };
934:     };
935:     };
936:     };
937:     };
938:     };
939:     };
940:     };
941:     };
942:     };
943:     };
944:     };
945:     };
946:     };
947:     };
948:     };
949:     };
950:     };
951:     };
952:     };
953:     };
954:     };
955:     };
956:     };
957:     };
958:     };
959:     };
960:     };
961:     };
962:     };
963:     };
964:     };
965:     };
966:     };
967:     };
968:     };
969:     };
970:     };
971:     };
972:     };
973:     };
974:     };
975:     };
976:     };
977:     };
978:     };
979:     };
980:     };
981:     };
982:     };
983:     };
984:     };
985:     };
986:     };
987:     };
988:     };
989:     };
990:     };
991:     };
992:     };
993:     };
994:     };
995:     };
996:     };
997:     };
998:     };
999:     };
1000:     };
1001:     };
1002:     };
1003:     };
1004:     };
1005:     };
1006:     };
1007:     };
1008:     };
1009:     };
1010:     };
1011:     };
1012:     };
1013:     };
1014:     };
1015:     };
1016:     };
1017:     };
1018:     };
1019:     };
1020:     };
1021:     };
1022:     };
1023:     };
1024:     };
1025:     };
1026:     };
1027:     };
1028:     };
1029:     };
1030:     };
1031:     };
1032:     };
1033:     };
1034:     };
1035:     };
1036:     };
1037:     };
1038:     };
1039:     };
1040:     };
1041:     };
1042:     };
1043:     };
1044:     };
1045:     };
1046:     };
1047:     };
1048:     };
1049:     };
1050:     };
1051:     };
1052:     };
1053:     };
1054:     };
1055:     };
1056:     };
1057:     };
1058:     };
1059:     };
1060:     };
1061:     };
1062:     };
1063:     };
1064:     };
1065:     };
1066:     };
1067:     };
1068:     };
1069:     };
1070:     };
1071:     };
1072:     };
1073:     };
1074:     };
1075:     };
1076:     };
1077:     };
1078:     };
1079:     };
1080:     };
1081:     };
1082:     };
1083:     };
1084:     };
1085:     };
1086:     };
1087:     };
1088:     };
1089:     };
1090:     };
1091:     };
1092:     };
1093:     };
1094:     };
1095:     };
1096:     };
1097:     };
1098:     };
1099:     };
1100:     };
1101:     };
1102:     };
1103:     };
1104:     };
1105:     };
1106:     };
1107:     };
1108:     };
1109:     };
1110:     };
1111:     };
1112:     };
1113:     };
1114:     };
1115:     };
1116:     };
1117:     };
1118:     };
1119:     };
1120:     };
1121:     };
1122:     };
1123:     };
1124:     };
1125:     };
1126:     };
1127:     };
1128:     };
1129:     };
1130:     };
1131:     };
1132:     };
1133:     };
1134:     };
1135:     };
1136:     };
1137:     };
1138:     };
1139:     };
1140:     };
1141:     };
1142:     };
1143:     };
1144:     };
1145:     };
1146:     };
1147:     };
1148:     };
1149:     };
1150:     };
1151:     };
1152:     };
1153:     };
1154:     };
1155:     };
1156:     };
1157:     };
1158:     };
1159:     };
1160:     };
1161:     };
1162:     };
1163:     };
1164:     };
1165:     };
1166:     };
1167:     };
1168:     };
1169:     };
1170:     };
1171:     };
1172:     };
1173:     };
1174:     };
1175:     };
1176:     };
1177:     };
1178:     };
1179:     };
1180:     };
1181:     };
1182:     };
1183:     };
1184:     };
1185:     };
1186:     };
1187:     };
1188:     };
1189:     };
1190:     };
1191:     };
1192:     };
1193:     };
1194:     };
1195:     };
1196:     };
1197:     };
1198:     };
1199:     };
1200:     };
1201:     };
1202:     };
1203:     };
1204:     };
1205:     };
1206:     };
1207:     };
1208:     };
1209:     };
1210:     };
1211:     };
1212:     };
1213:     };
1214:     };
1215:     };
1216:     };
1217:     };
1218:     };
1219:     };
1220:     };
1221:     };
1222:     };
1223:     };
1224:     };
1225:     };
1226:     };
1227:     };
1228:     };
1229:     };
1230:     };
1231:     };
1232:     };
1233:     };
1234:     };
1235:     };
1236:     };
1237:     };
1238:     };
1239:     };
1240:     };
1241:     };
1242:     };
1243:     };
1244:     };
1245:     };
1246:     };
1247:     };
1248:     };
1249:     };
1250:     };
1251:     };
1252:     };
1253:     };
1254:     };
1255:     };
1256:     };
1257:     };
1258:     };
1259:     };
1260:     };
1261:     };
1262:     };
1263:     };
1264:     };
1265:     };
1266:     };
1267:     };
1268:     };
1269:     };
1270:     };
1271:     };
1272:     };
1273:     };
1274:     };
1275:     };
1276:     };
1277:     };
1278:     };
1279:     };
1280:     };
1281:     };
1282:     };
1283:     };
1284:     };
1285:     };
1286:     };
1287:     };
1288:
```

```

131: pr_NONE, "PrinterControlWindow", 0, {}, 0, 0, 1,
132: pr_NONE, "-----", 0, {}, 0, 0, 1,
133: pr_NONE, "1990 by G. Glendon", 0, {}, 0, 0, 1,
134: pr_COMMAND, "Reset Printer", 0, {FS, 'e'}, 2, 1, 0,
135: pr_EXTERN, "----HELP----", 0, {}, 2, 3, 1,
136: pr_EXTERN, "Q U I T", 0, {}, 1, 2, 3
137: /* Extern-Routine #2 */
138: pr_EXTERN, "    "
139: /* Extern-Routine #1 */
140: );
141:
142: #define NUM_MENS 7
143:
144: struct GfxBase *GfxBase;
145: struct IntuitionBase *IntuitionBase;
146: struct RastPort *rp;
147: struct FileHandle *fh;
148: struct IntuiMessage *msg;
149: struct Window *win;
150:
151: struct Gadget *gadgad;
152: int gid;
153: long class;
154: int code; /* IntuiMsg-Decode */
155: int max; /* Maximale Anzahl Menü-Seiten */
156: struct prt *men; /* Zeiger auf aktuelle Menü-Liste */
157:
158: char Beep[2] = {7, 7}; /* Für Drucker */
159:
160: int blk = -1, nb = 0, slct; /* Aktueller 6er-Block, neuer 6er-Block
161: /* selektierte Funktion */
162:
163: main()
164: {
165: int temp, slt;
166: long cl;
167: int keypressed = 0;
168: OpenStuff(); /* Libs & Window öffnen */
169: men = menu; /* Menü : Hauptmenü */
170: max = NUM_MENS;
171: for (;;) {
172: if (nb != blk) {
173: /* Seitennummer hat sich geändert,
174: /* Menü muß neu angezeigt werden. */
175: blk = nb;
176: Display_em();
177: WaitPort(win->UserPort);
178: /* Damit keine Busy-Wait Schleife:
179: /* Task ist im Status 'Wait' und
180: /* verbraucht keine Prozessor-Zeit */
181: msg = GetMsg(win->UserPort);
182: class = msg->Class;
183: code = msg->Code;
184: gadgad = (struct Gadget *)msg->IAddress;
185: ReplyMsg(msg);
186:
187: switch (class) {
188: case RAWKEY:
189:
190: if (code > 127) keypressed = 0;
191: if (!keypressed) {
192: if ((code > 0) && (code < 7))
193: DoIt(blk * 6 + code - 1, keypressed = 1);
194: else
195: switch (code) {
196: case 76: /* Cursor up */
197: case 77: /* Cursor dn */
198: keypressed = 1;
199: code = 2;
200: nb = blk + code - 153;
201: if (nb > max) nb = 0;
202: if (nb < 0) nb = max - 1;
203: break;
204: case 95: /* Help */
205: men = info;
206: keypressed = 1;
207: max = NUM_INFO;
208: blk = -1;
209: nb = 0;
210: break;
211: case 16:
212: keypressed = 1;
213: ex(1);
214: break;
215: }
216: }
217: break;
218: case INACTIVEWINDOW:
219: if (win->Height > 86)
220: SizeWindow(win, 0L, -76L);
221: break;
222: case ACTIVEWINDOW:
223: if (win->Height < 88) {
224: /* Warten, bis linke Maustaste los-
225: /* gelassen wird.
226: /* Anmerkung: Könnte weggelassen
227: /* werden, wenn Intuition die
228: /* ACTIVEW.-Message erst nach dem
229: /* Loslassen der Drag-Bar schicken
230: /* würde ...
231: *asm
232: move.l d0, -(a7)
233: move.b $bfe001, d0
234: and.b $40, d0
235: beq.s lbl
236: move.l (a7)+, d0
237: *endasm
238:
239: WaitIT(3);
240: temp =
241: win->WScreen->Height - 88 - win->TopEdge;
242: if (temp < 0) {
243: /* Window ist zu weit unten um es
244: /* zu vergrößern, also noch oben
245: /* schieben...
246: *
247:
248:
249:
250:
251:
252:

```

Listing: Printer Control Window V2.0

```

253: WaitIT(5);
254: MoveWindow(win, 0L, (long)temp);
255: WaitIT(5);
256: }
257: SizeWindow(win, 0L, 76L);
258: WaitIT(5);
259: RefreshGadgets(&Gadget1, win, NL);
260: Display_em();
261:
262: break;
263: case GADGETUP:
264: gid = gadgad->GadgetID;
265:
266: if (gid < 30) {
267: /* Einer der normalen Menüpunkte */
268: slct = blk * 6 + gid - 21;
269: DoIt(slct);
270: }
271:
272: else {
273: /* einer der Pfeile oder 'Info' */
274: gid = 31;
275: switch (gid) {
276: case -1:
277: case 1: /* Pfeile : */
278: nb = blk + gid;
279: if (nb > max) nb = 0;
280: if (nb < 0) nb = max - 1;
281: break;
282: case 0: /* Info : */
283: men = info;
284: max = NUM_INFO;
285: blk = -1;
286: nb = 0;
287: break;
288: }
289: break;
290: }
291: }
292: }
293: }
294:
295: VOID *OpenLibrary();
296:
297: OpenStuff()
298: {
299: int t;
300: MakeImages();
301: if (!IntuitionBase =
302: OpenLibrary("intuition.library", 0L))
303: ex(0);
304: /* Anmerkung: Falls Intuition nicht
305: /* geöffnet werden konnte, kann auch
306: /* kein Requester gebracht werden! */
307:
308: if (!GfxBase =
309: OpenLibrary("graphics.library", 0L))
310: Error(UBP"Wo ist Graphics ?", ex(0);
311: /* Graphics */
312:
313: if (!win = OpenWindow(&windef))
314: Error(UBP"Hilfe: Kein Window:", ex(0);
315: /* und das Window */
316:
317: rp = win->RPort; /* RastPort holen und vorbereiten */
318: SetDrMd(rp, JAM2);
319: SetAPen(rp, 1L);
320: SetBPen(rp, 0L);
321:
322: if (!fh = Open("par:", MODE_NEWFILE))
323: Error(UBP"Drucker belegt ?", ex(0);
324:
325: Close(fh);
326:
327: /* Jetzt ist das 'parallel.device'
328: /* vorerst resident und muß nicht
329: /* mehr von der Sys.-Disk geladen
330: /* werden...
331: */
332:
333: /* Text-Längen eintragen : */
334: for (t = 0; t < (NUM_MENS * 6); t++)
335: menu[t].Len = (BYTE)strlen(menu[t].Name);
336: for (t = 0; t < (NUM_INFO * 6); t++)
337: info[t].Len = (BYTE)strlen(info[t].Name);
338: }
339:
340: ex(v)
341: int v;
342: /* Schließen aller Libraries und des Windows,
343: /* Verlassen des Programms
344: /* Wenn v=0, dann ohne Requester, ansonsten mit.
345: {
346: if (v) v = Error(UBP"Programm verlassen ?");
347: if (!v) {
348: if (win) CloseWindow(win);
349: if (GfxBase) CloseLibrary(GfxBase);
350: if (IntuitionBase) CloseLibrary(IntuitionBase);
351: RemoveImages();
352: Exit(0L);
353: }
354: }
355:
356: DoIt(num)
357: int num;
358: {
359: switch (men[num].Type) {
360: case pr_COMMAND:
361: /* Es wurde ein Drucker-Befehl
362: /* ausgewählt
363: if (!fh = Open("par:", MODE_NEWFILE))
364: Error(UBP"Drucker belegt ?", ex(0);
365: Write(fh, &(men[num].Prt), (long)(men[num].PLen));
366: Write(fh, Beep, 1L);
367: Close(fh);
368: break;
369:
370: case pr_EXTERN:
371: /* Es wurde ein Extern-Befehl ange-
372: /* wählt. Hier wird je nach Nummer
373:

```

Listing: Printer Control Window V2.0



```

374:         (in Plen) dekodiert, welche
375:         Funktion ausgewählt wurde. */
376:         switch (men[num].Plen) {
377:             case 1: /* Quit */
378:                 ex(1);
379:                 break;
380:             case 2: /* Info */
381:                 men=info;
382:                 max=NUM_INFO;
383:                 blk=-1;
384:                 nb=0;
385:                 break;
386:             case 3: /* norm. Menü */
387:                 men=menu;
388:                 max=NUM_MENS;
389:                 blk=-1;
390:                 nb=0;
391:                 break;
392:         }
393:     }
394: }
395:
396: /* Diese Routine zeigt die aktuelle Menüseite an. Dabei
397: wird vorher der Ausgabebereich gelöscht. */
398:
399: Display_em()
400: {
401:     int t,b,n;
402:     char p[MAXLINE+2];
403:     rp=win->RPort;
404:     b=blk*6;
405:     SetAPen(rp,0L);
406:     RectFill(rp,9L,14L,173L,83L);
407:     for (t=0;t<6;t++) {
408:         for (n=0;n<MAXLINE;p[n++]=' ');
409:         strcpy(p,men[b+t].Name);
410:         SetAPen(rp,(long)(men[b+t].FPen));
411:         SetBPen(rp,(long)(men[b+t].BPen));
412:         Move(rp,12L,(long)((t*12)+21));
413:         Text(rp,men[b+t].Name,(long)strlen(men[b+t].Name));
414:     }
415: }
416:
417: /* Dieser Teil erzeugt einen AutoRequest mit dem Text,
418: den man beim Aufruf übergeben hat. */
419:
420: struct IntuiText IText3 = {0,1,JAM2,20,26,&TOPAZ80,
421:     NL,NL}; /* wird vom Programm eingesetzt */
422: struct IntuiText IText2 = {0,1,JAM2,20,13,&TOPAZ80,
423:     USBP" Bitte bestätigen:",&ITex3};
424: struct IntuiText IText1 = {0,1,JAM2,20,5,&TOPAZ80,
425:     USBP" A C H T U N G :&ITex2};
426: struct IntuiText OKt = {0,1,JAM2,6,3,&TOPAZ80,USBP"OK",NL};
427: struct IntuiText NEINT = {0,1,JAM2,6,3,&TOPAZ80,USBP"Nein",NL};
428:
429: int Error(txt)
430: UBYTE *txt;
431: {
432:     IText3.IText=txt;
433:     return(AutoRequest(win,&ITex1,&NEINT,&OKt,
434:         NL,NL,272L,69L));
435: }
436:
437: MakeImages()
438: /* Kopiert Images ins Chip-Ram und biegt Zeiger um */
439: {
440:     ID1=AllocMem(28L,MEMF_CHIP);
441:     ID2=AllocMem(28L,MEMF_CHIP);
442:     ID3=AllocMem(28L,MEMF_CHIP);
443:     if (!(ID4=AllocMem(28L,MEMF_CHIP)))
444:         Error("Nicht genug Chip-Memory");
445:     Error("Gadgets können def. sein!");
446:     /* Eigentlich müßte bei jedem
447:     AllocMem() der erfolgreiche Ab-
448:     schluß überprüft werden, aber
449:     wenn der letzte OK ist, müßte der
450:     auch in Ordnung sein ( in 99.9%
451:     der Fälle...) */
452:     CopyIm(&ImageData2,ID2);
453:     CopyIm(&ImageData3,ID3);
454:
455:     CopyIm(&ImageData4,ID4);
456:     CopyIm(&ImageData1,ID1);
457:     Image1.ImageData=ID1;
458:     Image2.ImageData=ID2;
459:     Image3.ImageData=ID3;
460:     Image4.ImageData=ID4;
461: }
462:
463: RemoveImages()
464: {
465:     /* Gibt den belegten Speicher an das System zurück.
466:     Hier sollte man auf jeden Fall prüfen, ob der Spei-
467:     cher überhaupt geholt wurde, ansonsten bekommt man
468:     einen netten kleinen Guru... (für IDn=NULL!) */
469:     if (ID1) FreeMem(ID1,28L);
470:     if (ID2) FreeMem(ID2,28L);
471:     if (ID3) FreeMem(ID3,28L);
472:     if (ID4) FreeMem(ID4,28L);
473: }
474:
475: CopyIm(src,dest)
476: UBYTE *src,*dest;
477: {
478:     int t;
479:     for (t=0;t<28;t++) *dest++=*src++;
480: }
481:
482: WaitIT(n)
483: int n;
484: {
485:     /* Wartet auf n Intuiticks. */
486: }

```

Listing: Printer Control Window V2.0

```

494: struct IntuiMessage *msg,*GetMsg();
495: ULONG IDCMP_old,c1;
496: IDCMP_Old=win->IDCMPFlags;
497: /* Alte Flags sichern */
498: ModifyIDCMP(win,IDCMP_old;INTUITICKS);
499: /* und neue aktivieren. */
500: for (;n;) {
501:     WaitPort(win->UserPort);
502:     msg=GetMsg(win->UserPort);
503:     c1=msg->Class;
504:     ReplyMsg(msg);
505:     if (c1==INTUITICKS) n--;
506: }
507: ModifyIDCMP(win,IDCMP_old);
508: /* Alte Flags holen */
509: for (;msg;) {
510:     /* Überflüssige Msgs abholen */
511:     msg=GetMsg(win->UserPort);
512:     if (msg) ReplyMsg(msg);
513: }
514: }
515: /***** Ende von prcw.c *****/

```

```

1: /***** pcw_defs.h *****/
2: /*
3: Include-File für Printer Control Window
4: /* (c) 1990 Garry Glendown, & AMIGA DOS
5: /* Sprache : Aztec C
6: *****/
7:
8: #define NL NULL
9:
10: SHORT BorderVectors1[] = {0,0,195,0,195,4,0,4,0,0};
11: SHORT BorderVectors2[] = {0,0,18,0,18,38,0,38,0,0};
12: struct Border Border1 = {-2,-1,3,0,JAM1,5,BorderVectors1,NL};
13: struct Border Border2 = {-2,-1,3,0,JAM1,5,BorderVectors2,NL};
14:
15: struct TextAttr TOPAZ80 =
16: {((STRPTR)"topaz.font",TOPAZ_EIGHTY,0,0);
17: struct IntuiText IText4 =
18: {1,0,JAM2,4,28,&TOPAZ80,(UBYTE *)"O",NL};
19: struct IntuiText IText3 =
20: {1,0,JAM2,4,19,&TOPAZ80,(UBYTE *)"F",&IText4};
21: struct IntuiText IText2 =
22: {1,0,JAM2,4,10,&TOPAZ80,(UBYTE *)"N",&IText3};
23: struct IntuiText IText1 =
24: {1,0,JAM2,4,1,&TOPAZ80,(UBYTE *)"I",&IText2};
25:
26: USHORT ImageData1[] =
27: {0xF83E,0xFB8E,0xFB8E,0x8BA2,0x739C,0xB8BA,0xDFF6,
28: 0xEFE,0xF7DE,0xFB8E,0xF7DE,0xEFE,0xFFFE,0xFFFE};
29: USHORT ImageData2[] =
30: {0xFB8E,0xF93E,0xF83E,0x8BA2,0x0380,0x0380,0x739C,
31: 0xB8BA,0xDFF6,0xEFE,0xF7DE,0xFB8E,0xF7DE,0xEFE};
32: USHORT ImageData3[] =
33: {0xFFFE,0xFFFE,0xEFE,0xF7DE,0xFB8E,0xF7DE,0xEFE,
34: 0xDFF6,0xB8BA,0x739C,0x8BA2,0xFB8E,0xFB8E,0xF83E};
35: USHORT ImageData4[] =
36: {0xEFE,0xF7DE,0xFB8E,0xF7DE,0xEFE,0xDFF6,0xB8BA,
37: 0x739C,0x0380,0x0380,0x8BA2,0xF83E,0xF93E,0xFB8E};
38:
39: USHORT *ID1,*ID2,*ID3,*ID4,*AllocMem();
40:
41: struct Image Image1 = {0,0,15,14,1,ImageData1,1,0,NL};
42: struct Image Image2 = {0,0,15,14,1,ImageData2,1,0,NL};
43: struct Image Image3 = {0,0,15,14,1,ImageData3,1,0,NL};
44: struct Image Image4 = {0,0,15,14,1,ImageData4,1,0,NL};
45:
46: struct Gadget Gadget10 =
47: {NL,9,86,192,3,GADGBOX:GADGHIMAGE,RELVERIFY,
48: BOOLGADGET,{APTR}&Border1,NL,NL,40,NL};
49: struct Gadget Gadget9 =
50: {NL,&Gadget10,188,30,15,37,NL,RELVERIFY,BOOLGADGET,
51: GET,{APTR}&Border2,NL,&IText1,NL,NL,31,NL};
52: struct Gadget Gadget8 =
53: {&Gadget9,9,74,173,9,NL,RELVERIFY,BOOLGADGET,
54: NL,NL,NL,NL,NL,26,NL};
55: struct Gadget Gadget7 =
56: {&Gadget8,9,52,173,9,NL,RELVERIFY,BOOLGADGET,
57: NL,NL,NL,NL,NL,25,NL};
58: struct Gadget Gadget6 =
59: {&Gadget7,9,50,173,9,NL,RELVERIFY,BOOLGADGET,
60: NL,NL,NL,NL,NL,24,NL};
61: struct Gadget Gadget5 =
62: {&Gadget6,9,38,173,9,NL,RELVERIFY,BOOLGADGET,
63: NL,NL,NL,NL,NL,23,NL};
64:
65: struct Gadget Gadget4 =
66: {&Gadget5,9,26,173,9,NL,RELVERIFY,BOOLGADGET,
67: NL,NL,NL,NL,NL,22,NL};
68: struct Gadget Gadget3 =
69: {&Gadget4,9,14,173,9,NL,RELVERIFY,BOOLGADGET,
70: NL,NL,NL,NL,NL,21,NL};
71: struct Gadget Gadget2 =
72: {&Gadget3,188,69,15,14,GADGHIMAGE:GADGIMAGE,RELVER
73: IFY,BOOLGADGET,{APTR}&Image1,{APTR}&Image2,NL,NL,NL,32
74: ,NL};
75: struct Gadget Gadget1 =
76: {&Gadget2,188,14,15,14,GADGHIMAGE:GADGIMAGE,RELVER
77: IFY,BOOLGADGET,{APTR}&Image3,{APTR}&Image4,NL,NL,NL,30
78: ,NL};
79:
80: struct NewWindow windef = {233,57,207,88,0,1,
81: GADGETUP:ACTIVEWINDOW;INACTIVEWINDOW:RAWKEY,
82: WINDOWDRAG;WINDOWDEPTH;RMBTRAP,&Gadget1,NL,
83: (UBYTE *)"PCW by G.Glendown",NL,NL,0,0,0,0,
84: WBENCHSCREEN};
85:
86: /***** Ende von pcw_defs.h *****/

```

Listing: Printer Control Window V2.0

Edgar Meyzis

# Aufsteigerwissen Assembler

In der (zunächst?) letzten Folge unseres Assemblerkurses wollen wir auf die Hardware-Ebene hinabsteigen, um Zugang zu einem Bereich zu gewinnen, für den Assembler am besten geeignet ist. Wir werden einzelne Bildpunkte setzen, daraus eine Linie komponieren und schließlich Herrn Blitter "en personne" auftreten lassen.

Unser Rückblick soll der vierten Folge gelten, in der wir Ihnen die Aufgabe stellten, zwei ähnliche Routinen zusammenzufassen. Die Lösung kann sehr unterschiedlich erfolgen.

## Ein Blick zurück

Getreu unserem Kursmotto haben wir einen einfachen, leicht nachvollziehbaren Weg gewählt, wie in Listing 1 und 2 dargestellt. Wir kopieren unsere Quelldatei "O4.asm" (Kurzform für ProjektMenu.asm) nach "O5.asm", entfernen daraus die Routinen "OpenBildDatei", "CloseBildDatei" sowie "BildSchreiben" und bauen "BildLesen" entsprechend Listing 1 aus. Weiterhin ist die Routine "Menu0Auswerten" entsprechend Listing 2 zu ändern.

Die Arbeitsweise unserer neuen Routine "BildTransferieren" steuern wir mit einem Parameter, den wir in Register d2 übergeben. Alternativ hätten wir auch über den Stack gehen können. Das Register d2 wurde gewählt, weil es gleich beim Öffnen der Datei als Parameterregister ohnehin gefordert ist. Es bot sich an, die Konstanten "modeOldFile" und "modeNewFile" zur Steuerung von "BildTransferieren" zu verwenden. Die neue Routine ist gegenüber dem bisherigen Vorgehen robuster geworden. Sie berücksichtigt, daß die Schreib- und Leseroutinen des "DOS" nicht immer ungestört ablaufen (Beispiel: Diskette ist voll beschrieben). "DOS" gibt im Fehlerfall einen negativen Wert (in d0) zurück, auf den mit "bmi.s \e" (branch minus) reagiert wird. Durch eine ausführliche Fehlerbehandlung einschließlich

einer Ausgabe der Ursache könnte die Routine verbessert werden. In der vorliegenden Form wird die Datei geschlossen (sofern vorher geöffnet) und so getan, als sei nichts geschehen. Das Vorgehen bewahrt vor "Abstürzen", ist aber benutzerunfreundlich.

Es soll noch auf den Assemblerbefehl "movem.l" eingegangen werden, den wir häufig eingesetzt haben, ohne zu betrachten, in welcher Reihenfolge die Register auf dem Stack abgelegt bzw. von dort geladen werden. Um es kurz zu machen, die Reihenfolge der Register im Quelltext ist bedeutungslos. Der Assembler wertet unsere Notation aus und setzt im zweiten Wort des Maschinenbefehls, das die Registermaske bildet, für jedes angesprochene Register ein bestimmtes Bit (ein Wort = 16 Bits = 8 Adreß- und 8 Datenregister der CPU). Die Registermaske bestimmt so-

Kursfahrplan	
Teil 1. Einführung und Grundlagen	
Teil 2. Vertiefung der Programmieretechnik	
Teil 3. Praktische Arbeit mit Graphics	
Teil 4. Praktische Arbeit mit DOS	
Teil 5. Hardwarenahe Programmierung von Grafik	

mit die Reihenfolge der Register auf dem Stack. Diese Feststellung läßt sich verallgemeinern: Mit der "movem.l" Instruktion werden Registerinhalte so im Arbeitsspeicher (also auch auf dem Stack) abgelegt, daß an der untersten Adresse die Datenregister, mit d0 beginnend, und anschließend die Adreßregister (a0 zuerst) Platz für ihren Inhalt finden. Mit zunehmender Speicheradresse steigt somit auch die Registernummer, beginnend bei d0 und endend bei a7. Im Quelltext nicht benannte Register werden einfach übergangen.

Bei einem Stack ist zu bedenken, daß er von oben nach unten wächst. Für den konkreten Fall unserer Routine "BildTransferieren" ergibt sich: Unabhängig von der Reihenfolge der Nennung der Register in der Anweisung "movem.l a2/d7/d2,-(sp)" weist "sp" (Stack Pointer) auf den Anfang der Speicherstellen, die den Inhalt von d2 aufgenommen haben.

Ohne in die Arbeitsweise des Blitters tief einzusteigen (das

überlassen wir unserer "Blitter-Ecke", die sich des Themas seit Ausgabe 2/90 annimmt), wollen wir nun "hardwarenäher" programmieren, um Grafikausgaben zu beschleunigen. Einen krönenden Abschluß muß eine strapaziöse Wüstendurchquerung schließlich finden. Der Kreis schließt sich:

## Fülle Ei mit Dampf

Heute wollen wir das Ei mit Dampf füllen, ich meine die Abarbeitung der Routine "FuelleEi" beschleunigen. Mit Rücksicht auf die hinzugekommenen Leser ist "FuelleEi" in Listing 3 wiedergegeben, allerdings in überarbeiteter Form, da wir zwischenzeitlich unsere Programmieretechnik verfeinert haben. Erstellen Sie anhand des Listings die Textdatei "G5.asm". Wir werden damit weiterarbeiten, um einige Funktionen des Blitters zu demonstrieren, die Sie bei Bedarf in Ihre "PaintBox" übernehmen können. Die Routine "AreaBlitterDemo", die sich zunächst nur um "FuelleEi" kümmert, rufen wir menügesteuert aus unserem Hauptprogramm "P5" auf. Sie können dafür ein weiteres Menü oder einen Menüpunkt anlegen. Die Technik dürfte Ihnen mittlerweile geläufig sein. Wir hingegen erweitern in "Z5" (vormals "Z4") die Routine "ZeichneBox" um den Aufruf "bsr AreaBlitterDemo". Noch haben wir nicht an "speed" zugelegt. Wir wollen zunächst nur die Gewißheit gewinnen, daß die Routine fehlerfrei arbeitet, um einen sicheren Ansatzpunkt zu schaffen.

So, nun kommt Dampf ins Ei, wie Sie Listing 4 entnehmen können. Bei dem bisherigen Vorgehen haben wir dem Blitter die Arbeitsbedingungen vorenthalten, unter denen er erst richtig zu packen kann. Da fehlt uns zunächst die

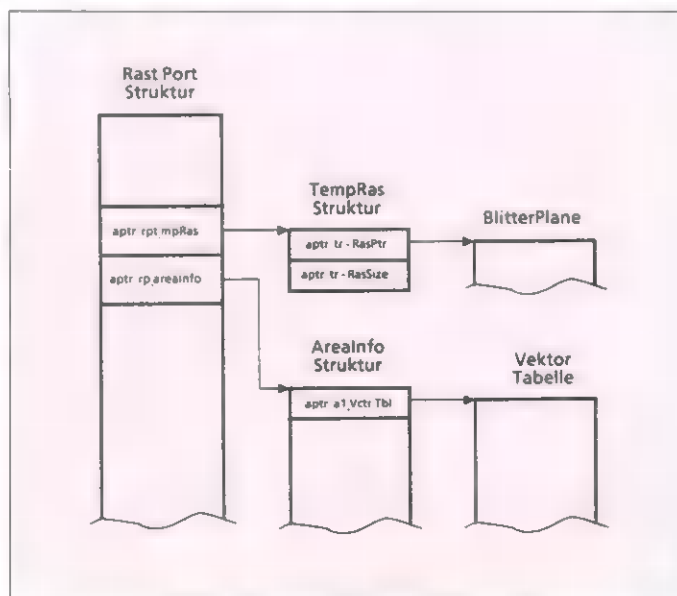


Bild 1. Die Routine "AreaOperation Vorbereiten" unterstützt unseren "Rastport" mit benötigten Datenstrukturen



Spielwiese für den Blitter, die Area, die wir mit der Systemroutine "AllocRaster" in der Größe unseres Bildschirms (640 \* 256) (Routine "BlitterPlaneAnlegen") schaffen. Weiterhin ist eine Kurzbeschreibung dieser "Blitter-Plane" mit der Systemroutine "InitTempRas" in einen Zwischenspeicher (Routine "TempRasAnlegen") zu schreiben und dessen Adresse in der Rast-Port-Struktur zu vermerken. Bild 1 stellt die Zusammenhänge dar. Angeforderter Arbeitsspeicher ist an das Betriebssystem zurückzumelden (mit "FreeRaster"), wenn nicht mehr benötigt. Das erledigt "BlitterPlaneAufgeben".

Wir packen das "Ei" quasi in ein Sandwich, bestehend aus den Routinen "AreaOperationenVorbereiten" und "AreaOperationenAbschliessen", um endlich in den vollen Genuß des Blitters zu gelangen. Ergänzen Sie bitte Ihr Listing 3 durch Listing 4. Na, wie gefällt Ihnen das? Sie können das Makro "FlaecheFuell" nun auch mit dem Füllmodus "1" aufrufen, um die Möglichkeit zu nutzen, beliebig begrenzte Flächen zu füllen und nicht einfach alles zu übertünchen, was innerhalb des Bereichs liegt, der mit dem "AreaOutlinePen" markiert ist. Die Ellipse ist fast gleichberechtigt. Aber es kommt noch besser! In Listing 4 haben wir bereits zwei "leere" Routinen ("AreaInfoAnlegen" und "AreaInfoAufgeben") vorgesehen, die wir im nächsten Abschnitt mit Code füllen werden.

Aus den Makros "FlaecheFuell" und den "AreaOperation"-Routinen könnten Sie Ihrer "PaintBox" ein Geschenk entwickeln, um beliebige Flächen zu füllen. Der Füllmodus sollte per Menü wählbar sein, ebenso die Funktion "Füllen". Letztere sollte die Position der Maus feststellen, wenn Sie den linken Knopf drücken, und die Fläche in der Farbe des Vordergrundstiftes (APen) füllen. Als Muster für das Vorgehen könnte die bewährte Routine "FreeDraw" dienen.

## Noch fetziger und mit Mustern

Bisher haben wir nur die "Flood-Funktion" der "Blitterorientierten" Systemroutinen genutzt. Nun gehen wir einen Schritt weiter und lassen den Blitter für uns geometrische

Gebilde zeichnen sowie die dabei entstehenden Flächen füllen. Dazu müssen wir noch eine "AreaInfo-Struktur" gemäß Listing 5 anlegen und deren Adresse in die Rast-Port-Struktur eintragen. Weiterhin ist es erforderlich, für die zu zeichnenden Figuren eine Tabelle anzulegen, in die die Routine "AreaDraw" Zeichenvektoren für den Blitter vermerkt. Der Zusammenhang zwischen den Strukturen "RastPort", "TempRas", "AreaInfo" und der Vektorentabelle geht aus Bild 1 hervor. Reservierte Speicherbereiche sind nach Abschluß der "Area-Operationen" an das Betriebssystem zurückzumelden. Den angelegten Strukturen können die jeweiligen Adressen entnommen werden, so daß es überflüssig ist, globale Variablen einzurichten. Auf die Reihenfolge der Rückmeldungen ist zu achten, um nicht die Gurus hinter der nächsten Sanddüne herauszufordern. "AreaInfoAufgeben" ist eine der Routinen, die Speicherplatz zurückmeldet. Sie wäre robuster, wenn sie zunächst prüfen würde, ob die Adressen der Speicherbereiche überhaupt gültig sind. Mit "AreaInfoAnlegen" und "AreaInfoAufgeben" haben wir nun alle Voraussetzungen geschaffen, mit "Areas", das heißt mit dem Blitter, zu arbeiten. Uns steht nun die uneingeschränkte Anwendung der Systemroutinen

- AreaMove (Startpunkt eines Vielecks bestimmen),
- AreaDraw (Eckpunkte des Polygons festlegen),
- AreaEllipse (Ellipse definieren),
- AreaEnd (mit zuvor genannten Befehlen bestimmte Figuren nun endlich zeichnen und ausfüllen).

offen. Die Sprungvektoren sind in Listing 5 vermerkt. Eine Demonstration der neuen Befehle bietet die Routine "Polygon-UndEllipse".

Beliebige Flächen können wir nun einfarbig füllen. Die Polygontechnik ließe sich einfach in die PaintBox integrieren. Auf unserer letzten Wüstenetappe wollen wir nun ein paar Spuren im Sand hinterlassen (besser als Coladosen), das heißt Flächen mit Mustern füllen. Auch das ist mit dem Blitter leicht möglich.

Listing 6 enthält alle Informationen über die Verwendung einfarbiger Muster. Das Makro "SetAFPt" (Set Area Fill Pattern) meldet das Muster bei der Rast-Port-Struktur an. Von nun an wird mit dem Muster gefüllt. Wir haben "Muster0" global deklariert. Muster haben stets eine Breite von 16 Bits, begrenzt durch die Auslegung des Blitters. Bei breiteren Flächen wiederholt sich das Muster. Die Höhe ist beliebig wählbar, jedoch nur in Schritten von Zweierpotenzen. Als Höhe ist nicht die Anzahl der Wörter, sondern der Exponent der Zweierpotenz anzugeben (in unserem Fall bei acht Wörtern somit drei).

Es lassen sich auch mehrfarbige Muster einfach erzeugen. Für jede Bitplane ist dann ein Muster anzulegen. Die Höhe des Musters ist wie im ersten Fall für eine Bitplane anzugeben, jedoch mit negativem Vorzeichen. Ergänzen Sie Listing 6 entsprechend. Sie werden von den Möglichkeiten begeistert sein.

## Superschnelle Punkte

Mit dem Blitter haben wir schon sehr nahe an der Hardware gearbeitet.

Mit Listing 7 stellen wir Ihnen Routinen vor, die Punkte setzen. Betrachten wir zunächst "SetzePunkt": Als erster Parameter ist die Adresse der Bitmap-Struktur anzugeben, auf die in der Rast-Port-Struktur verwiesen wird. In "Graphics.i" finden Sie die Vereinbarung der Bitmap-Struktur. Sie gibt Auskunft über die Größe und die Tiefe (das heißt Anzahl der Bitplanes) eines RastPort. Weiterhin enthält Sie ab "bm\_planes" ein Feld mit den Adressen der einzelnen Bitplanes. Die Adressen benötigen wir, um in den Planes Punkte zu setzen. Als Parameter für "SetzePunkt" wäre der aktuelle RastPort gleichfalls geeignet.

Gehen wir nun die Anweisungen von "SetzePunkt" durch. Zunächst holen wir uns die Adresse der ersten Bitplane. Dann ermitteln wir, in welchem Byte einer Bildschirmzeile die X-Position liegt. Die Multiplikation der Y-Position mit 80 (640 Bildpunkte geteilt durch acht) ergibt die Adresse der Zeile, bezogen auf den Anfang der Bitplane. (Die Multiplikation erfordert mehr Zeit als ein "shift #6,..." und ein "shift #4,..." sowie Addition der erhaltenen Werte. Aus Gründen der Anschaulichkeit haben wir auf diese Möglichkeit der Optimierung verzichtet.) Die Addition der Zeilenadresse mit der Adresse des Bytes innerhalb der Zeile ergibt schließlich die Adresse des Bytes, bezogen auf den jeweiligen Anfang unserer vier Bitplanes, in dem wir ein Bit, das heißt unseren Punkt, zu setzen haben.

Die Adresse des Punktes innerhalb des Bytes erhalten wir durch die drei Anweisungen "and.w #7,d3 sub.q #7 and neg.w d3". Bei der ganzzahligen Multiplikation durch acht haben wir die unteren drei Bits verloren. Die holen wir uns mit der ersten Anweisung wieder und kennen nun die Position des zu setzenden Pixels innerhalb des Bytes. Leider ist noch eine Umformung des erhaltenen Ergebnisses notwendig, weil Bits von rechts nach links durchnumeriert werden, während die X-Positionen von links nach rechts zunehmen. Der aufgezeigten Gegenläufigkeit wird durch die beiden weiteren Anweisungen Rechnung getragen. Das Register d3 enthält schließlich die genaue Bezeichnung des Bits.

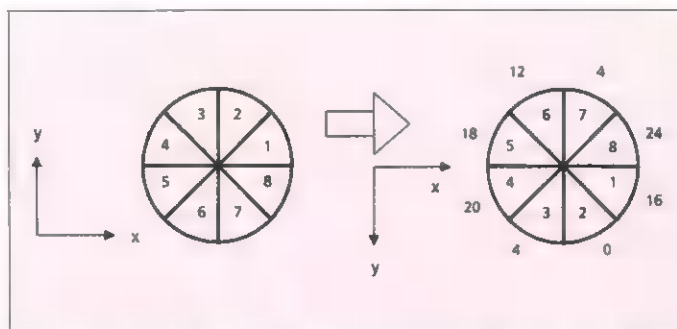


Bild 2. Durch Drehen der x-Achse passen wir die Oktantennumeration der Lage des Koordinatensystems der Amiga-Grafik an. Außerhalb des rechten Kreises: Oktantenbezeichnung nach Blitterart

Unmittelbar hinter dem Label "\L0" testen wir, ob in der aktuellen Bitplane ein Bit zu setzen ist. Die Bits null bis drei entsprechen unseren vier Bitplanes. Falls kein Bit zu setzen ist, verzweigen wir zu "\L1" und holen uns die Adresse der nächsten Bitplane. Dann schieben im Farbregister d2 die Bits nach rechts, um das mit der aktuellen Bitplane korrespondierende Bit an der Position null zu haben. Schließlich fangen wir wieder bei "\L0" an. Dieses Vorgehen ist sehr schnell. Es wird jedoch von dem Nachteil begleitet, daß es zu Farbverfälschungen kommen kann, weil nur Punkte hinzugefügt werden.

Wenn ein Punkt zu setzen ist, dann wird zu der Adresse der Bitplane (a1) die Adresse innerhalb der Plane (d1) addiert und das Bit gesetzt ("bset d3,(a1)"). Die Routine "SetzePunkt" hat sich in Simulationssystemen bestens bewährt.

In der "SetzePunktDemo" wird eine Gerade gezogen, deren Farbe von Punkt zu Punkt wechselt. Für jeden einzelnen Punkt der Gerade werden die Koordinaten und die Farbe an "SetzePunkt" übergeben. Ein ungeheurer Aufwand für eine Gerade. Im Fall einer Kurve sieht das aber schon ganz anders aus. Wie gefällt Ihnen die Geschwindigkeit der Routine?

## Linien mit dem Blitter

Einzelne Punkte können wir nun setzen, und auch für geschwungene Linien haben wir einen rasanten Weg gefunden. Nun fordern wir den Blitter unmittelbar auf, für uns Geraden zu ziehen. Es gilt zunächst den Oktanten (Achtelkreis) zu bestimmen, der die grobe Richtung der Linie beschreibt. Wo fängt man bei der Bezeichnung der Oktanten bei einem Kreis an? Konventionen besagen, daß bei "drei Uhr" der Oktant Null beginnt und daß entgegengesetzt dem Uhrzeigersinn durchnummeriert wird. Auf dem Amiga ist aber alles anders. Aus Gründen der Anschaulichkeit beugen wir uns der hier üblichen Konvention, die Oktanten von eins bis acht durchnummerieren. Am Anfang des Listing 8 finden Sie entsprechende Vereinbarungen, die in die Datei "incl/Hardware.i" eingehen sollten.

Wir könnten somit bei einer Bezeichnung der Oktanten gemäß der linken Hälfte von Bild 2 bleiben. Im Gegensatz zu den mathematischen Konventionen liegt beim Amiga der Nullpunkt links oben in der Ecke und nicht links unten. Es folgt: Ein zunehmendes Delta-Y ( $Y2 - Y1$ ) bedeutet auf dem Amiga keine positive, sondern eine negative Steigung. Aus Gründen der Anschaulichkeit ist es daher zweckmäßig, die Oktantenbezeichnung entsprechend Bild 2, rechter Teil, der Lage des Koordinatensystems des Amiga anzupassen.

Leider versteht der Blitter nicht unsere Numerierung der Oktanten. Er will anders informiert werden. Dazu dienen die Dezimalzahlen außerhalb des rechten Kreises (Bild 2). Sie finden die Werte auch in Listing 8 in der Tabelle "Oktanten" wieder. Die Berechnung von Oktanten ist wesentlich einfacher als häufig angenommen. In Bild 2 werden die notwendigen Überlegungen ausführlich dargestellt.

Mit Blick auf Bild 2, rechter Teil, stellen wir fest, daß der Endpunkt der Linie in den Oktanten 7, 8, 1 oder 2 liegen muß, wenn  $X1 < X2$  : Oktanten 3, 4, 5 oder 6).

Weiterhin gilt: Wenn  $Y1 < Y2$  ist, dann liegt der Endpunkt in den Oktanten 5, 6, 7 oder 8 ( $Y1 > Y2$  : Oktanten 4, 3, 2 oder 1).

Mit den bisherigen Folgerungen könnte man nur auf den Viertelkreis schließen, in dem der Endpunkt der Linie liegt. Eine weitere Bedingung hilft

uns, vom Viertelkreis zum Oktanten zu gelangen, indem wir die Steigung der Linie einbeziehen: Wenn der Absolutwert von Delta-X größer ist als der Absolutwert von Delta-Y, dann kommen nur die Oktanten 5, 8, 4 oder 1 in Frage; im umgekehrten Fall verbleiben die Oktanten 6, 7, 3 oder 2.

Durch Auswertung der drei Bedingungen läßt sich der gesuchte Oktant sicher bestimmen. Dazu könnte man zum Beispiel entsprechend Listing 10 verfahren. Wer mag schon die ineinander verwobenen Verzweigungen? Einem derartig formulierten Algorithmus mangelt es an Klarheit. An der Aussage ändert sich nicht viel, wenn man sicherstellt, daß  $X1$  stets größer als  $X2$  ist. Durch Vertauschen ließe sich das bewerkstelligen, und der Blitter hätte auch nichts dagegen. Die möglichen Lösungen würden dann immer im rechten Halbkreis liegen.

Wir gehen einen etwas anderen Weg und wenden in Listing 8 ein Stück Mengenlehre an, das wir in Listing 9 noch ein wenig verfeinern. Dazu legen wir uns eine Tabelle mit den Oktanten an. Vergleichen Sie bitte die drei formulierten Bedingungen mit der Tabelle, die mit dem Label "X1\_kl\_X2" beginnt. Erkennen Sie die Abbildung der hergeleiteten Bedingungen. Wir wollen entsprechend der Ergebnisse der drei Vergleiche die drei Zeilen der Tabelle bestimmen, die den gesuchten Oktanten enthalten. Wir wissen, daß der gesuchte Oktant in allen drei Zeilen vor-

kommt. Jede Zeile ist 32 Bits lang. Folglich muß eine UND-Verknüpfung (Schnittmenge) der drei Zeilen den gesuchten Oktanten zum Ergebnis haben. Genau diese Aufgabe leistet der erste Teil der Routine (Listing 8) bis zum Label "\Linie".

Warum muß eigentlich ein Langwort mit acht Halbbytes (Nibbles) herhalten, um die Oktanten aufzunehmen? Ein Byte müßte doch als Datenstruktur ausreichen, da es gleichfalls acht verschiedene Werte annehmen kann? Sehr richtig! In Listing 9 führen wir auch gleich diese Verfeinerung ein. Der Code wird insgesamt kürzer und im Schnitt auch schneller.

Verfolgen wir nun Listing 8 ab "\Linie" weiter, nachdem uns der erste Teil so beschäftigt hat: Wir haben genauso wie für "SetzePunkt" die Adresse des Bytes zu bestimmen, in dem der Bildpunkt liegt. Auch die Position des Pixel innerhalb des Bytes ist gleichermaßen zu bestimmen. Am Schluß der Routine, vor dem Label "\wart" treffen wir auch auf die uns schon geläufige Technik, die Adresse der nächsten Bitplane zu erlangen.

Auf die eigentliche Versorgung des Blitters mit Daten wollen wir hier nicht eingehen; das macht die "Blitter-Ecke". Abweichend zu "SetzePunkt" benutzen wir ein Zählregister, um die vier Bitplanes zu bearbeiten. Weiterhin löschen wir in einer Bitplane die Bits an der Stelle der Linie, die nicht zu ihrer Farbe passen, sie somit umfärben würden. Das Löschen hätten wir durch Ändern der Maske in "bltdata(a1)" auf Null erledigen können. Eleganter erscheint uns jedoch der Wechsel der "Minterms" von #SFA auf #SA, zumal die Möglichkeit erhalten bleibt, gemusterte Linien zu ziehen. In Bild 3 haben wir den Versuch unternommen darzustellen, wie die drei möglichen Signalquellen durch den Blitter im Linienmodus verarbeitet werden. Als Quellen dienen:

1. die blitterintern erzeugte Anweisung, ein Bit zu setzen (oder auch nicht), Kanal A,
2. die Maske mit dem Linienmuster, Kanal B, der wir schon früher begegnet sind und
3. der Inhalt der Bitplane, Kanal C, die zu bearbeiten ist.

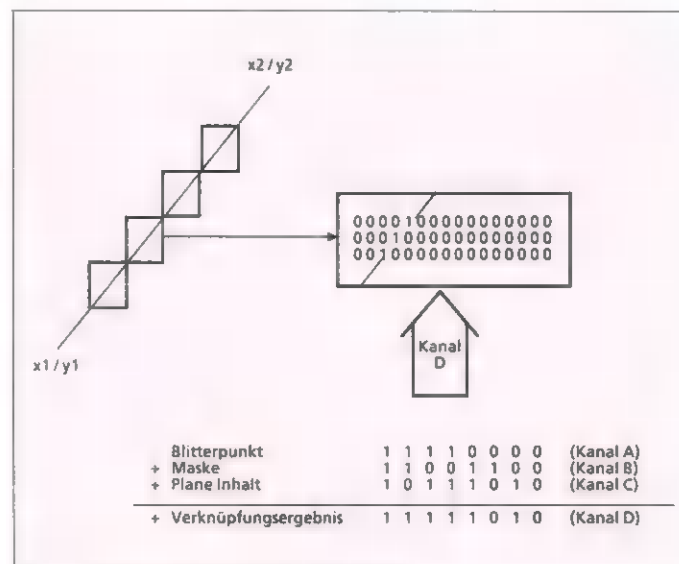


Bild 3. Der Blitter läßt den Drachen steigen



Die Bitplane ist somit Quelle (Kanal C) und Ziel (Kanal D) zugleich. Die MinTerms (kurze boolsche Ausdrücke) bestimmen, wie die Informationen der Kanäle zu verknüpfen sind.

Beispiel: Die zweite Spalte von links ist wie folgt zu interpretieren: Wenn der Blitter intern berechnet hat, daß ein Punkt zu setzen ist (Blitterpunkt = 1), und die Maske an der Bitposition des Punktes eine Eins enthält (Maske = 1) und an der entsprechenden Stelle der Bitplane kein Punkt gesetzt ist, dann soll der Blitter tun, was er meint, nämlich den Punkt setzen.

Beachten Sie bitte, daß Sie die Zeile mit den Verknüpfungsergebnissen selbst bestimmen. Durch Eintragen von Null oder Eins legen Sie fest, wie der Blitter zu arbeiten hat. Bedenken Sie bitte auch, daß der Blitter nicht punktweise, sondern zeilenweise arbeitet und somit das von Ihnen vorgegebene "Verknüpfungsverhalten" (Wortschöpfung!) jeweils für die Breite eines Wortes gilt. Mit der Vorgabe legen Sie eigentlich die Verknüpfungsregeln fest. Die Reihenfolge der Spalten ist so (unveränderlich) festgelegt, wie in Bild 3 dargestellt [1]. Das erwartete Verhalten des Blitters

(Verknüpfungsergebnis) läßt sich sedezimal ausdrücken. Im dargestellten Beispiel ergibt sich #SFA.

Prüfen Sie es in Listing 8 nach: Wir füttern den Blitter mit dem Wert mit den Anweisungen "or.w #SFA, con0" (kurz nach \Li- nie) und "move.b #SFA, con0" (kurz nach "\warte"). Ist in der jeweiligen Bitplane kein Punkt zu setzen, dann beschränken wir uns auf #SA (unmittelbar vor \n). Vergleichen Sie unser Verhalten mit Bild 3. Im letzten Fall begnügen wir uns mit der Übernahme bereits gesetzter Bildpunkte, wie im rechten Teil des Bildes dargestellt. Sollten

Sie dem Blitter bei der Arbeit zuschauen wollen, dann ersetzen Sie bitte #SFA vorübergehend durch #SA0.

## Ziel erreicht

Das Ende der heutigen Etappe bedeutet zugleich, daß wir das Ende des Kurses erreicht haben. Es hat Freude gemacht, für Sie zu schreiben. Teilen Sie uns mit, wenn Sie Interesse an einer ähnlichen Tour verspüren. Warum nicht einmal durch das Packeis der Arktis?

(mm)

## Listings

```

1:
2: BildTransferieren ; Parameter in d2
3: *-----
4: * #modeOldFile : Bild einlesen
5: * #modeNewFile : Bild speichern
6:
7:     movem.l a2/d7/d2,-(sp) ; Arbreg + Param sich.
8:
9:     move.l gadgBuffer(locals),d1 ; zu oeffnende Datei
10:    CallD0s Open ; Mode bereits in d2
11:    move.l d0,bildHdl(locals) ; FileHandle oder null
12:    beq.s \f ; falls nicht geklappt
13:
14:    move.w #3,d7 ; vier Bitplanes
15:    move.l Screen(globals),a2 ; Zeiger auf Screen
16:
17:    * in Screen-Struktur ist Bitmap-Struktur eingelagert,
18:    * die ab bm_Planes, die Adressen unserer Bitplanes enthaelt
19:
20:    lea.l sc_bitMap+bm_Planes(a2),a2 ; auf PlaneSpeich.
21:    move.l (a2),d2 ; dieselbe holen
22:    \speicherLoop
23:
24:    * Jede Zeile enthaelt 640 Bildpunkte, d.h. 80 Bytes
25:    * Unser Screen ist 256 Bildpunkte hoch
26:
27:    move.l #80*256,d3 ; Anzahl Bytes pro
28:    Plane $5000
29:    move.l bildHdl(locals),d1 ; von Datei
30:
31:    cmp.l #modeNewFile,(sp) ; Bild speichern ?
32:    beq.s \w ; ja
33:    CallD0s Read ; 1 Plane einlesen
34:    bra.s \x ; zur Fehlerpruefung
35:    \w CallD0s Write ; 1 Plane speichern
36:    \x bml.s \e ; falls Fehler aufgetreten
37:
38:    adda.l #4,a2 ; auf naechste Plane-
39:    Speicher-Adr
40:    move.l (a2),d2 ; die Plane-Adr holen
41:    dbra d7,\speicherLoop ; und Lesen
42:
43:    \e move.l bildHdl(locals),d1 ; FileHandle Bilddatei
44:    beq.s \f ; falls ungueltig
45:    CallD0s Close
46:    \f movem.l (sp)+,a2/d7/d2
47:    rts

```

Listing 1: Loesung (Teil 1) zur Aufgabe aus Folge 4

Durch Parametrisierung lassen sich die Schreib- und Leseroutinen leicht zusammenfassen

```

1:
2: .....
3:
4:     tst.w ItemNr(locals); Laden oder Sichern ?
5:     bne.s \s
6:     move.l #modeOldFile,d2 ; Parameteruebergabe in d2
7:     bra.s \bt
8:     \s move.l #modeNewFile,d2 ; Sichern
9:     \bt
10:    bar BildTransferieren
11:    \c1
12:    move.l (sp)+,a3
13:
14:    .....

```

Listing 2: Teil 2 der Loesung zur Aufgabe aus der letzten Folge  
Die Routine "MenuAuswerten" ist wie dargestellt zu aendern, um "BildTransferieren" aufzurufen.

```

1: # Grafikspielereien (G5.asm)
2:
3: include "incl/Types.i"
4: include "incl/Graphics.i"

```

Listings zum Assemblerkurs

```

5: include "incl/Intuition.i"
6:
7: AllocRaster = -492 ; (width,height){d0/d1}
8: FreeRaster = -498 ; (planePtr,width,height){a0,d0/d1}
9: InitTempRas = -468 ; (tmpras,buffer,size){a0/a1,d0}
10:
11: * Export
12:
13: PUBLIC AreaBlitterDemo ; ausserhalb verfuegbar machen
14:
15: * Import
16:
17: PUBLIC GraphBase ; aus Hauptmodul
18: PUBLIC RastPort
19:
20: * Registergebrauch
21:
22: globals equir a5 ; Zeiger auf globale Variablen
23: our_win equir a4 ; Zeiger auf geoeffnetes Fenster
24:
25:
26:
27: DoRpfunct macro ; Grafikfunktion in einem
28: *----- Rastport (Rp) ausfuehren
29: * Param: Funktion
30:     move.l RastPort(globals),a1 ; Rastport-Adr holen
31:     CallGraph \1 ; Funktion ausfuehren
32:     endm
33:
34:
35: SetzePen macro ; Param: XPos, YPos
36: *----- Rastport (Rp) ausfuehren
37:     move.w \1,d0 ; XPos in Parameterreg.
38:     move.w \2,d1 ; YPos in Parameterreg.
39:     DoRpfunct Move ; Stiftposition setzen
40:     endm
41:
42: FaerbeAPen macro ; Para: Farbreghisternummer
43: *----- ; Farbe ins Parameterregister
44:     move.w \1,d0
45:     DoRpfunct SetAPen
46:     endm
47:
48: SetOpen macro ; Para: Farbnummer
49: *----- ; Rastport-Adresse
50:     move.l RastPort(globals),a1 ; Rastport-Adresse
51:     move.b \1,rp_a0Pen(a1) ; Outlinefarbe
52:     in Rp aendern
53:     endm
54:
55: Ellipse macro ; Para: XM, YM, Xr, Yr
56: *-----
57:     move.w \1,d0
58:     move.w \2,d1
59:     move.w \3,d2
60:     move.w \4,d3
61:     DoRpfunct DrawEllipse
62:     endm
63:
64:
65: Flaechefuell macro
66: *----- Para: Mode, XPos, YPos
67:     move.w \1,d2 ; Fuell Mode, 0 oder 1
68:     move.w \2,d0 ; XPos
69:     move.w \3,d1 ; YPos
70:     DoRpfunct Flood
71:     endm
72:
73:
74: FuelleE1
75: *-----
76:     FaerbeAPen #13
77:     Ellipse #200,#110,#200,#100
78:     SetOpen #13
79:     Flaechefuell #0,#200,#110
80:     rts
81:
82: AreaBlitterDemo
83: *-----
84:     bar FuelleE1
85:     rts
86:
87:     end

```

Listing 3: Ohne volle Nutzung des Blitters werden die Flaechen sehr langsam gefueellt.

Listings zum Assemblerkurs

```

1: include "incl/Types.i"
2: include "incl/Graphics.i"
3: include "incl/Intuition.i"
4: include "incl/Exec.i"
5:
6: AllocRaster = -492 ; (width,height)(d0/d1)
7: FreeRaster = -498 ; (planePtr,width,height)(a0,d0/d1)
8: InitTempRas = -468 ; (tmpRas,buffer,size)(a0/a1,d0)
9:
10: * TempRas
11: struktur
12:   aptr tr_RasPtr
13:   aptr tr_RasSize
14:   struktLaenge tr_size
15:
16:
17: BlitterPlaneAnlegen
18: *-----
19:   move.w wd_Width(our_Win),d0 ; Pixel/Zeile
20:   move.w wd_Height(our_Win),d1 ; Bildhoehe
21:   CallGraph AllocRaster ; Adr BlitterPlane
22:   rts ; in d0
23:
24:
25: BlitterPlaneAufgeben ; Parameter
26: *-----
27:   move.w wd_Width(our_Win),d0 ; a0 : Adr BlitterPlane
28:   move.w wd_Height(our_Win),d1 ; Pixel/Zeile
29:   CallGraph FreeRaster ; Pixel/Spalte
30:   rts
31:
32:
33: TempRasAnlegen ; Parameter: a1 Adr BlitterPlane
34: *-----
35:   move.l a2,-(sp) ; Ergebnis in d0
36:   move.l a1,a2
37:   AllocMemory #tr_size,#Mem_Clear ; TempRas-Struktur
38:   tst.l d0 ; angelegt ?
39:   beq.s \f ; falls nicht
40:   move.l d0,a0 ; Adr TempRas-Struktur
41:   move.l a0,-(sp) ; Adr TempRas merken
42:   clr.l d0
43:   move.w wd_Width(our_Win),d0 ; Pixel/Zeile
44:   divu #8,d0 ; Byte/Zeile
45:   mulu wd_Height(our_Win),d0 ; Byte/Plane
46:   move.l a2,a1
47:   CallGraph InitTempRas
48:
49:   move.l wd_RPort(our_Win),a0 ; Rastport Adr holen und
50:   move.l (sp),rp_tmpRas(a0) ; Adr TempRas eintragen
51:   move.l (sp)+,d0 ; Adr TempRas
52:   \f
53:   move.l (sp)+,a2
54:   rts
55:
56:
57: TempRasAufgeben
58: *-----
59:   move.l a2,-(sp)
60:   move.l RastPort(globals),a2 ; aus RastPort
61:   FreeMemory rp_tmpRas(a2),#tr_size ; tempRas aufgeben
62:   clr.l rp_tmpRas(a2) ; in RastPort vermerken
63:   move.l (sp)+,a2
64:   rts
65:
66:
67: AreaInfoAnlegen
68: *-----
69:   rts
70:
71: AreaInfoAufgeben
72: *-----
73:   rts
74:
75:
76: AreaOperationenVorbereiten ; ohne Parameter, Ergebnis
77:   in d0
78:   *----- Ergebnis:Adr BlitterPl.
79:   o. null
80:   move.l d2,-(sp)
81:   bsr BlitterPlaneAnlegen
82:   move.l d0,d2 ; Adr BlitterPlane merken
83:   beq.s \z ; keine BlitterPlane da
84:   bsr TempRasAnlegen
85:   tst.l d0 ; angelegt + eingetragen ?
86:   beq.s \x ; nein
87:   move.l #20,d0 ; 20 Koordinatenpaare
88:   bsr AreaInfoAnlegen
89:   tst.l d0 ; angelegt + eingetragen ?
90:   bne.s \z ; ja
91:   bsr TempRasAufgeben ; da kein AreaInfo verfueg
92:   bar
93:   \x
94:   move.l d2,a0 ; Adr BlitterPlane
95:   bsr BlitterPlaneAufgeben ; Speicher zurueck
96:   clr.l (sp) ; Fehler signalisieren
97:   \z
98:   move.l d2,d0 ; Ergebnis holen
99:   move.l (sp)+,d2
100:   rts
101:
102:
103: AreaOperationenAbschliessen
104: *-----
105:   move.l RastPort(globals),a0 ; aus RastPort
106:   move.l rp_tmpRas(a0),a0 ; ueber TempRas
107:   move.l tr_RasPtr(a0),a0 ; Adr BlitterPlane holen
108:   bsr BlitterPlaneAufgeben
109:   bsr TempRasAufgeben
110:   bsr AreaInfoAufgeben
111:   rts
112:
113:
114:
115:
116:
117:
118:
119:
120:
121:
122:
123:
124:
125:
126:
127:
128:
129:
130:
131:
132:
133:
134:
135:
136:
137:
138:
139:
140:
141:
142:

```

Listings zum Assemblerkurs

```

119: SetOpen #13
120: FlaechenFuell #0,#200,#110
121:
122: Beispiel2 ; Fuelle Flaechen im Mod. 1
123: FaerbaAPen #9 ; mit Farbe 9
124: Ellipse #200,#80,#70,#60
125: FlaechenFuell #1,#200,#80
126: X
127: FaerbaAPen #12
128: FlaechenFuell #1,#200,#30
129:
130: rts
131:
132:
133:
134:
135:
136:
137:
138:
139:
140:
141:
142:

```

Listing 4: Gebt dem Blitter seine Spielwiese, damit er Flaechen blitzschnell fuellen kann.

```

1:
2: InitArea = -282 ; (AreaInfo,vectorTable,vectorTable
3:   Size)
4: *
5:   AreaDraw = -258 ; (RastPort,XPos,YPos)(a1,d0/d1)
6:   AreaEllipse = -186 ; (RastPort,XPos,YPos,a,b)(a1,d0-d3)
7:   AreaEnd = -264 ; (RastPort),(a1)
8:   AreaMove = -252 ; (RastPort,XPos,YPos)(a1,d0/d1)
9:
10: * Area Info ; mit dieser Struktur arbeitet der
11:   struktur ; Blitter bei allen Area-
12:   Operationen
13:   aptr ai_VctrTbl ; Sie wird mit InitArea
14:   initialisiert
15:   aptr ai_VctrPtr ; Wichtig ist es, die
16:   VektorTabelle ; ausreichend zu dimensionieren
17:   aptr ai_FlagTbl ; Auf die Tabelle verweist
18:   ai_VctrTbl
19:   word ai_Count
20:   word ai_MaxCount
21:   word ai_FirstX
22:   word ai_FirstY
23:   struktLaenge ai_size
24:
25:
26: AreaInfoAnlegen ; Param: d0: Anzahl Koordinatenpaare
27:   *----- Ergebnis in d0 (<0 erfolgreich)
28:   move.l d2/d3/a2/a3,-(sp)
29:   move.l d0,d3 ; Anz Koordinatenpaare aufbewahren
30:   AllocMemory #ai_size,#Mem_Clear ; AreaInfo-Struktur
31:   tst.l d0 ; Struktur angelegt ?
32:   beq.s Fehler ; nein
33:   move.l RastPort(globals),a3 ; Zeiger auf RastPort
34:   move.l d0,rp_areaInfo(a3) ; Struktur eintragen
35:   move.l d0,a2 ; Zeiger auf AreaInfo
36:
37:   move.l d3,d2 ; Anz Koordinatenpaare
38:   mulu #5,d2 ; 5 Bytes pro Paar
39:   AllocMemory d2,#Mem_Clear ; Vektortabelle anlegen
40:   tst.l d0 ; Tabelle angelegt
41:   bne.s Weiter ; ja
42:
43:   FreeMemory a2,#ai_size ; AreaInfo aufgeben
44:   clr.l rp_areaInfo(a3) ; in RastPort vermerken
45:   clr.l d0 ; Fehler signalisieren
46:   bra.s Fehler
47:
48:   Weiter
49:   move.l d0,a1 ; Adr Vektortabelle
50:   move.l a2,a0 ; Adr AreaInfo
51:   move.l d3,d0 ; Anz Koordinatenpaare
52:   CallGraph InitArea
53:   Fehler
54:   move.l (sp)+,d2/d3/a2/a3
55:   rts
56:
57:
58: AreaInfoAufgeben
59: *-----
60:   move.l a2,-(sp)
61:   move.l RastPort(globals),a2 ; RastPort holen
62:   move.l rp_areaInfo(a2),a2 ; Zeiger auf AreaInfo
63:
64:   move.l #5,d1 ; 5 Bytes/Vektor
65:
66:   mulu ai_MaxCount(a2),d1 ; Vektortab. in Bytes
67:
68:   FreeMemory ai_VctrTbl(a2),d1 ; Tabelle aufgeben
69:
70:   FreeMemory a2,#ai_size ; AreaInfo aufgeben
71:   clr.l rp_areaInfo(a2) ; in RastPort vermerken
72:   move.l (sp)+,a2
73:   rts
74:
75:
76:
77:
78:
79:
80:
81:
82:
83:
84:
85:
86:

```

Listings zum Assemblerkurs



```

87: move.l a2,a1 ; 2. Vektor
88: move.l #380,d0
89: move.l #200,d1
90: CallGraph AreaDraw
91:
92: move.l a2,a1 ; 3. Vektor
93: move.l #340,d0 ; Ausgangspunkt gleich
94: move.l #140,d1 ; Endpunkt wird selbst
95: CallGraph AreaDraw ; erkannt
96:
97: move.l a2,a1 ; RastPort
98: move.l #100,d0 ; Mittelpunk
99: move.l #140,d1
100: move.l #70,d2 ; Radien
101: move.l #70,d3
102: CallGraph AreaEllipse
103:
104: FaerbeApen #14 ; Farbe fuer beide Figuren
105: move.l a2,a1
106: CallGraph AreaEnd ; beide Figuren zeichnen
107: movem.l (sp)+,a2/d2-d3
108: rts

```

Listing 5: Fuer die Arbeit mit den Area-Routinen bedarf es einer initialisierten AreaInfo-Struktur und einer Vektortabelle. Der Aufwand wird durch schnellste Grafik belohnt. Eine Kostprobe bietet "PolygonUndEllipse".

```

1:
2: SetAfPt macro ; RastPort mit Füllpattern versehen
3: -----
4: ; Params: /1 Adr RastPort, /2 Adr Muster
5: ; /3 Musterhoehe (Exponent, Zweierpotenz)
6: move.l \1,a1 ; RastPort holen
7: move.l \2,ra_areaPtrn(a1) ; Muster eintragen
8: move.b \3,ra_areaPtSz(a1) ; Musterhoehe eintragen
9: endm
10:
11: ; so ist SetAfPt einzusetzen
12: move.l RastPort(globals),a1
13: lea.l Muster0(pc),a0
14: SetAfPt a1,a0,#3 ; Muster bei RastPort anmelden
15: bsr PolygonUndEllipse ; Muster anwenden
16:
17: SetAfPt #0,#0,#0 ; Muster zuruecksetzen
18:
19:
20: Muster0 dc.w %1100110011001100 ; als Füllmuster
21: dc.w %0110011001100110 ; es sind 2*3 Woerter

```

Listings zum Assemblerkurs

```

22: dc.w %0011001100110011 ; somit ist 3 der
23: dc.w $FF ; Parameter fuer die
24: dc.w $FF ; Hoehe des Musters
25: dc.w %0011001100110011
26: dc.w %0110011001100110
27: dc.w %1100110011001100

```

Listing 6: "SetAfPt" fuert die RastPort-Struktur mit den Informationen ueber ein Pattern, das bei Fülloperationen zu verwenden ist.

```

1:
2: SetzePunkt | Parameter: a0 BitMap-Adr
3: -----
4: ; d0 XPos
5: ; d1 YPos
6: ; d2 Farbe
7: lea.l bm_planes(a0),a0 ; erste BitPlane
8: move.w d0,d3 ; XPos merken
9: lsr.w #3,d0 ; XPos/8 = Byte der Zeile
10:
11: mulu #80,d1 ; Zeilenaddr. in BitPlane
12: add.w d0,d1 ; Adr Byte in dem Punkt liegt
13: and.w #7,d3 ; Pos. des Punktes im Byte
14:
15: subq.w #7,d3 ; BitNr ermitteln
16: neg.w d3 ; BitNr ist stets positiv
17:
18: and.w #$F,d2 ; auf 16 Farben begrenzen
19: ; entspr. 4 BitPlanes
20: \0
21: btst #0,d2 ; Punkt in aktueller Plane
22: ; zu setzen?
23: beq.s \L1 ; nein
24:
25: move.l (a0),a1 ; aktuelle Plane holen
26: adda.l d1,a1 ; ergibt absolute Zeilenaddr.
27: bset d3,(a1) ; Punkt setzen
28:
29: \L1
30: addq.l #4,a0 ; auf naechste BitPlane
31: lsr.b #1,d2 ; naechste Farbebene nach Bit 0
32: tst.b d2 ; noch Farbebenen zu bearbeit. ?
33: bne.s \0 ; ja
34: rts
35:
36: SetzePunktDemo
37: -----
38: movem.l a2/d2-d7,-(sp)
39: move.l #600,d4 ; StartPosition X
40: move.l #200,d5 ; Y

```

Listings zum Assemblerkurs

2 MB nur 698,-

DATA 2000 GmbH + Co. KG 5800 HAGEN 1  
 Stresemannstraße 11-16, Tel. 02331/23290 + 31272  
 Fax. 23231. Lieferung per Nachnahme oder Vorkasse  
 + 10,- + 1,50 Versch. Mo.-Fr. 10.00-18.30 Sa. bis 14/16.00

DATA 2000

<b>Amiga DMA-Portexpander 98,-</b> <ul style="list-style-type: none"> <li>für Amiga 500/1000</li> <li>mit dieser Karte ist es möglich, bis zu 5 weitere Karten parallel zu betreiben</li> <li>jeder Port einzeln schaltbar</li> <li>angesprochene Karte wird optisch über LED angezeigt</li> <li>86-polige Messerleisten (männlich) werden mitgeliefert</li> <li>somit Einsatz von Stecker u. Buchsen, incl. Stützfüße</li> </ul>	<b>Amiga Speichererweiterung 512K 198,-</b> <ul style="list-style-type: none"> <li>für Amiga</li> <li>Gasat-Speicher 1</li> <li>fest aufgebaute Platine</li> <li>kein Eingriff in den Rechner</li> </ul>	<b>Amiga Epromkarte 1 MB 129,-</b> <ul style="list-style-type: none"> <li>für Amiga 500/1000</li> <li>Alternative zur Floppy, schnell wie eine RAM-Floppy, anzusprechen mit dir rom</li> <li>Steuersoftware auf Disk, auch Nachladeprogramme können geladen werden</li> <li>für Epromtypen 27512 und 27010</li> </ul>	<b>Amiga Sound-Sampler 79,-</b> <ul style="list-style-type: none"> <li>für Amiga 500/2000</li> <li>Audio-Genie, Profi-Perfect-Sound</li> <li>Digitalisierung rauscharm, für Sprache und Musik, Anschluß am Druckerport</li> <li>Steckanschlüsse in Cinch</li> <li>eingebauter Vorverstärker</li> <li>Software auf Diskette</li> </ul>	<b>Interne-Kick-Um</b> <ul style="list-style-type: none"> <li>für A500/2000 "B"</li> <li>für 2 x ROM</li> <li>und 8 x Eprom</li> <li>mit Schalter</li> <li>Kick-Um1 79,-</li> <li>ROM 1.2 49,-</li> <li>ROM 1.3 59,-</li> <li>dfo, für A2000 "A"</li> <li>Kick-Um2 79,-</li> <li>Bootselektoren</li> <li>A500/1000 BI 18,-</li> <li>A2000 mit LW 18,-</li> </ul>	<b>Amiga DMA-Portadapter 29,-</b> <ul style="list-style-type: none"> <li>für Amiga 500/1000</li> <li>DMA-Port wird verlängert</li> </ul>
<b>NEUHEIT Amiga-Light-Mouse 98,-</b> Nachdem der Amiga-Lightpen schon lange Zeit angekündigt war, haben wir jeglichen Kompromiß verworfen. Viele Lösungen waren für uns nicht professionell genug, z. B. wenn man mit einer Hand den Lightpen und mit der anderen die Mouse halten muß. Die Amiga-Light-Mouse hat beide Mauseinstellen bereits eingebaut. Spitzensoftware liegt auf Diskette bei.	<b>Amiga Relaiskarte 149,-</b> <ul style="list-style-type: none"> <li>für Amiga 500/1000</li> <li>8 Kanal/16 Kanal</li> <li>mit Steuerelectronic, II separat zu steuernde Relais</li> <li>je 1 x UM Kontakt, bis zu 220V/3A</li> <li>einschl. List</li> <li>Anschluß DMA-Port</li> <li>externe Anschlüsse über Kleinleisten</li> <li>Betrieb an 5 V vom Rechner</li> <li>bei voller Nutzung Anschluß für externes Netzteil vorhanden (Steckernetzteil)</li> </ul>	<b>Amiga 3,5-Zoll-Floppy extern 229,-</b> <ul style="list-style-type: none"> <li>für alle Amiga</li> <li>durchgeführter BUS, abschaltbar</li> <li>1-Zoll-Metallgehäuse</li> </ul>	<b>Amiga-STEREO-Sound-Sampler 149,-</b> sonst wie oben, jedoch für A 500/1000/2000 <b>Solange Vorrat reicht</b> Sidecar ungeprüft 120,- Netzteil 120,- Floppy 90,- Gehäuse 90,- Mainboard + Interf. 149,-	<b>Externe-Kick-Um</b> <ul style="list-style-type: none"> <li>die erste unseres Wissens zum Anschluß an DMA-Port</li> <li>für Versionen</li> <li>z.B. 1.2 = 1.3 usw.</li> <li>auf 2 x 4 Eproms</li> <li>bei A1000 256 K mehr Speicher durch WORM-Einbindung</li> <li>mit Software auf Disk mit Kickstartmaster</li> <li>deutsche Beschreib.</li> <li>durchgeführter DMA Port für A500 98,-</li> <li>für A1000 98,-</li> </ul>	<b>Amiga DMA-Winkeladapter 39,-</b> <ul style="list-style-type: none"> <li>mit einer 90° Winkelabzweigung, also 2 Steckmöglichkeiten</li> </ul>
<b>Amiga Testboard 25,-</b> <ul style="list-style-type: none"> <li>für alle Amiga</li> <li>Anschluß für S-D-Stecker</li> <li>86polig, 2 x 43</li> </ul>	<b>Eprommer 149,-</b> <ul style="list-style-type: none"> <li>für alle Amiga</li> <li>liest, vergleicht, brennt Eproms, Proms, CMOS-Typen 2716-27011</li> <li>IK in 14 Sekunden</li> <li>Betrieb Druckerport</li> <li>3 Algorithmen wählbar, mit Textoolsocket, Software on Disk</li> <li>incl. Stützfüße</li> </ul>	<b>Interface + HD-Treiber-Software Auftr. i. A.L.F.-Software 98,-</b> <b>Amiga Userport Testboard 79,-</b> <ul style="list-style-type: none"> <li>für Amiga 500/1000</li> <li>incl. 2 x 6522, Userport am DMA</li> <li>Pio-Karte</li> <li>Testboard</li> <li>gepuffert</li> <li>Lochrasterkarte im Raster 2,54</li> <li>doppelseitig</li> </ul>	<b>Sonderpreis 598,-</b> <b>Interne-RAM-Karte</b> <ul style="list-style-type: none"> <li>neue Neuheit kommt pünktl. zum Herbst auf Markt</li> <li>mit Uhr + Akku</li> <li>paßt ins Bodenfach</li> <li>abschaltbar</li> <li>IRAM 1 fertige Karte f. 41256 od. 51100. Bitte Preis angeben. Geprüft ohne RAMs</li> <li>IRAM3 mit RAMs</li> <li>RAM 2 298,-</li> <li>auf 1 MB bestückt</li> <li>RAM 3 449,-</li> </ul>	<b>Externe-RAM-Karte</b> <ul style="list-style-type: none"> <li>film A500 + 1000</li> <li>als 0.5 MB-System</li> <li>als 1.0 MB-System</li> <li>als 0.5 relesistestes Kickstart-RAM-WORM</li> <li>oder gemischt 0.5 MB + 0.5 Kick usw.</li> <li>wahlweise 41256/Megabit</li> <li>durchgeführter DMA Port</li> </ul>	<b>Amiga Epromkarte 2 MB 159,-</b> siehe hierzu auch den Test in Amiga 12/89
<b>Amiga 69,-</b> <ul style="list-style-type: none"> <li>für Amiga 500/1000</li> <li>stufenloses Herunterregeln von Spielen und Programmen auf Null (durch Poti)</li> <li>Herstellen von Bildschirmfotos</li> </ul>	<b>Shugate-Interface 29,-</b> <ul style="list-style-type: none"> <li>für alle Amiga</li> <li>zum Anschluß von passenden 3,5-Zoll-Laufwerken an Amiga-Rechnern</li> <li>Kabelsatz im Lieferumfang</li> </ul>	<b>Amiga Midi-Interface 79,-</b> <ul style="list-style-type: none"> <li>für Amiga 500/2000</li> <li>1 x Midi in, 2 x Midi out, 1 x Midi out thru</li> <li>schaltbar, incl. Anschlußkabel</li> <li>Pilot-level</li> <li>für den Amiga 1000</li> <li>fehlen wir Wandler 9221</li> </ul>	<b>Amiga Harddisk</b> Amiga 500 20 MByte Speicherkapazität mit Form-Position inkl. Handbuch-Book und behandeln wie an 2. Laufwerk Software-Handgeleitet im separaten Gehäuse 32x24x2 cm fertig installiert, sofort einsatzbereit für jeden User bedienbar alle Kabel on Board angeschlossen und fertig	<b>ab Ende März '90 lieferbar</b> <b>RAM 1 fertige Karte, ohne RAMs</b> RAM 1 148,- RAM 2 mit RAMs auf 0.5 MB bestückt RAM 2 298,- auf 1 MB bestückt RAM 3 449,-	Diese Produkte sind auch erhältlich bei: Ali Moabit 106 1000 Berlin 21 Tel. 030/3925316 Weitzels Computerzubehör Tiefenstraße 8 4150 Krefeld Tel. 0215/2121150 Spezialelektronik Gießler Straße 7031 Leipzig Tel. 0037/41740144

```

40: move.l #150,d6 ; Farben
41: move.l d6,d7 ; Anzahl zu setzende Punkte
42: move.l RastPort(globals),a2
43: \
44: move.l rp_bitMap(a2),a0 ; Adr BitMap
45: move.w d4,d0 ; XPos
46: move.w d5,d1 ; YPos
47: move.w d6,d2 ; Farbe
48: bar SetzePunkt
49: subq.w #1,d4 ; neue XPos
50: subq.w #1,d5 ; neue YPos
51: subq.w #1,d6 ; neue Farbe
52: dbeq d7,\
53: movem.l (sp)+,a2/d2-d7
54: rts
55:

```

Listing 7: Superschnelle Punkte mit einer massgeschneider-  
ten Routine ohne Overhead, aber auch ohne  
Schutz vor Unbeabsichtigtem Ueberschreiben des  
Speichers.

```

1: # Die nachfolgenden Vereinbarungen gehoerten
2: # in eine Datei namens "incl/Hardware.l"
3:
4: custom = $0FFF000 ; Hardwareadresse
5: ; Custom-Chip
6: ; Offsets fuer Register
7: ; des Blitters
8: bitddat = $000
9: dmaconr = $002
10: bitcon0 = $040
11: bitcon1 = $042
12: bitafwm = $044
13: bitalwm = $046
14: bitcpt = $048
15: bitbpt = $04C
16: bitapt = $050
17: bitdpt = $054
18: bitsize = $058
19: bitcmmod = $060
20: bitbmod = $062
21: bitamod = $064
22: bitdmmod = $066
23: bitcdat = $070
24: bitbdat = $072
25: bitadat = $074
26: dmacon = $098
27:
28: LINEMODE = $001 ; versetzt Blitter in Linienmodus
29: SIGNFLAG = $040 ; bei negativen Steigungen benoetigt
30:
31: BLITTER = 06 ; Auskunft ueber Blitter-Status
32: BLTDONE = 14
33:
34: OCTANT1 = 16 ; Oktanten, blittergerecht bezeichnet
35: OCTANT2 = 00
36: OCTANT3 = 08
37: OCTANT4 = 20
38: OCTANT5 = 28
39: OCTANT6 = 12
40: OCTANT7 = 04
41: OCTANT8 = 24
42:
43: Ziehelinie ; mit dem Blitter
44: ----- Para: d0 = x1
45: ; d1 = y1
46: ; d2 = x2
47: ; d3 = y2
48: ; a3 = BitMap (mit 4 Planes)
49: ; d7 = Farbe
50:
51: x1 ecur d0
52: x2 ecur d2
53: y1 ecur d1
54: y2 ecur d3
55:
56: dx ecur d2
57: dy ecur d3
58:
59: # Oktant ermitteln zur Unterstuetzung des Blitters
60:
61: lea.l x1_k1_x2(pc),a0
62: sub.w x1,x2 ; x2 - x1 = dx
63: bpl.s \y ; dx positiv ?
64: addq.l #4,a0 ; dx naechste Oktantenzeile
65: neg.w dx ; falls dx negativ
66: \
67: lea.l y1_k1_y2(pc),a1
68: sub.w y1,y2 ; y2 - y1 = dy
69:
70: bpl.s \d
71: addq.l #4,a1
72: neg.w dy
73: \
74: lea.l DX_gr_DY(pc),a2
75: cmp.w dy,dx ; dx > dy ?
76: bpl.s \d1 ; ja
77: addq.l #4,a2
78: exg dx,dy ; Algorithmus erwartet dx > dy
79: \
80: move.l (a0),d5 ; Schnittmenge der 3 Oktantenmengen
81: and.l (a1),d5 ; bilden.
82: and.l (a2),d5 ; Ergebnis: Gesuchter Oktant
83: tat.w d5 ; Oktant im Langwort so verschieben,
84: bne.s \s ; dass er im unteren Nibble landet
85: swap d5
86: \
87: tat.b d5
88: bne.s \s1
89: asr.w #8,d5 ; Motorola Syntax : #0
90: \
91: cmp.b #F,d5
92: bml.s \s2
93: asr.b #4,d5 ; Oktant nun im unteren Nibble
94: lea.l Oktanten(pc),a0 ; Oktant als Offset verdoppeln
95: add.w d5,d1 ; da Tabelle wortweise angelegt
96: move.w 0(a0,d5.w),d5 ; Oktant blittergerecht holen
97: ; einschliesslich des Linemodus
98: \
99: Linie
100: move.l #80,d4 ; spezial. Routine auf Breite 640
101: mulu.w d4,y1 ; ergibt Zeilenaddr innerhalb Plane

```

Listings zum Assemblerkurs

```

99:
100: * x1 zerlegen in Byte und Pixel
101:
102: move.l x1,d6
103: lsr.w #3,d6 ; Nr. des Bytes in der Zeile
104: add.w d6,y1 ; Addr. des Bytes bezogen auf Plane
105: move.l y1,a2 ; fuer andere Planes merken
106: and.w #7,d0 ; Pixel innerhalb des Bytes
107:
108: con0 ecur d0
109:
110: ror.w #4,con0 ; obere 4 Bits enthalten Pixelpos.
111: or.w #BFA,con0 ; nun die DMA-Kanäle zuweisen
112: ; B40 zeigt Arbeitsart d. Blitt.,
113: ; Fenster fuer Fenster als Treppe
114: lsl.w #2,d3 ; 4 * dy
115: add.w d2,d2 ; 2 * dx
116:
117: blt ecur d1
118:
119: move.w d2,blt ; Wort Blitterstart zusammensetzen
120: lsl.w #5,blt ; 2 * dx (= Linienlaenge setzen)
121: add.w #B42,blt ; Bits u. 1 setzen (Hoehe,
122: ; Breite)
123: lea.l custom,a1 ; Blitter adressieren
124: btst #BLITTER,dmaconr(a1) ; empfohlener Testaufruf
125: \
126: btst #BLITTER,dmaconr(a1)
127: bne.s \ErwarteBlitter
128:
129: move.w #B400,dmaconr(a1) ; %1000010000000000
130:
131: ; = Blitter DMA Prio-
132: ; ritast ueber CPU
133: ; = Prioritaet setzen
134: move.w dy,bltbmod(a1) ; 4 * dy
135: sub.w d2,d3 ; 4 * dy - 2 * dx
136: ext.l d3
137: move.l d3,d6 ; Steigung fuer BLTAPT
138: bpl.s \w
139: or.w #SIGNFLAG,d5 ; zu setzen, wenn fallend
140: \
141: move.w d4,bitcmmod(a1) ; Modulo (Bildbreite), fuer
142: ; gerade und ungerade Planes
143: sub.w d2,d3 ; 4 * dy - 4 * dx
144: move.w d3,bitamod(a1) ; Steigung
145: move.l #FFFFFF,bitafwm(a1) ; First Word Mask
146: ; Kanal A
147: move.w #B000,bltadat(a1) ; fuer Linien alles 1
148: move.w #FFFF,bltbdat(a1) ; so im Line-Modus
149: ; gerade und ungerade Planes
150: move.w #3,d2 ; Maske auf 111...
151: ; 4 Bitplanes bearbeiten
152: \
153: loop
154: move.l (a3),a0 ; Addr Bitplane holen
155: add.l a2,a0 ; Byteaddr innerh. Plane dazu
156: \
157: btst #BLITTER,dmaconr(a1)
158: bne.s \warte
159:
160: move.b #FA,con0 ; Annahme Punkt setzen
161: btst #0,d7 ; Punkt in Plane setzen ?
162: bne.s \n ; ja
163: move.b #0A,con0 ; kein Punkt in dieser Plane
164: ; zu setzen aber loeschen
165: \
166: move.l d6,bltapt(a1)
167: move.l a0,bltcdat(a1) ; Linie, Anfangsaddr.
168: move.l a0,bltdpt(a1) ; Linie, Anfangsaddr.
169: move.w con0,bltcon0(a1) ; Anfangspos + DMA-Zugriff
170: move.w d5,bltcon1(a1) ;
171: move.w blt,bltsize(a1) ; Starte Blitter
172:
173: addq.l #4,a3 ; auf naechste Plane
174: lsr.b #1,d7 ; naechste Farbebene ins Bit 0
175: dbra.s d2,\loop
176:
177: \
178: btst #BLITTER,dmaconr(a1)
179: bne.s \warte
180:
181: move.w #B400,dmaconr(a1) ; %0000010000000000
182: ; = Blitter DMA Prio-
183: ; ritast ueber CPU
184: ; = Prioritaet aufheben
185: rts
186:
187: AreaBlitterDemo
188: -----
189: ...
190: movem.l d2-d7/a2-a5,-(sp)
191: move.l #20,d1 ; x1
192: move.l #160,d1 ; y1
193: move.l #600,d2 ; x2
194: move.l #40,d3 ; y2
195: move.w #8,d7 ; Farbe violett
196: move.w RastPort(globals),a2 ; Adr BitMap
197: move.l rp_bitMap(a2),a0 ; erste BitPlane
198: lea.l bm_planes(a0),a3
199:
200: bar Ziehelinie
201: movem.l (sp)+,d2-d7/a2-a5
202: rts
203:
204: cnop 0,4
205:
206: * Tabelle der Oktanten (1 - 8)
207: X1_k1_X2 dc.l $12000078
208: dc.l $00345600
209: Y1_k1_Y2 dc.l $12340000
210: dc.l $00005678
211: DX_gr_DY dc.l $10045008
212: dc.l $02300670
213:
214: * Tabelle zur blittergerechten Umwandlung
215: # der Oktanten bei gleichzeitiger Addition
216: # des LINEMODE
217: Oktanten dc.w 0 ; dummy

```

Listings zum Assemblerkurs



```

217:      dc.w OCTANT1+LINEMODE ; Zusammenschreiben!
218:      dc.w OCTANT2+LINEMODE
219:      dc.w OCTANT3+LINEMODE
220:      dc.w OCTANT4+LINEMODE
221:      dc.w OCTANT5+LINEMODE
222:      dc.w OCTANT6+LINEMODE
223:      dc.w OCTANT7+LINEMODE
224:      dc.w OCTANT8+LINEMODE
225:

```

Listing 8: Linien mit dem Blitter erfordern einen hohen Programmieraufwand. Die Muehe lohnt sich, wenn == auf die Ausfuehrungsgeschwindigkeit ankommt.

```

1:      # Oktant ermitteln zur Unterstuetzung des Blitters
2:
3:      lea.l x1_k1_X2(pc),a0
4:      sub.w x1,x2 ; x2 - x1 = dx
5:      bpl.s \y ; dx positiv?
6:      addq.l #1,a0 ; auf naechste Oktantenzeile
7:      neg.w dx ; falls dx negativ
8:      \y
9:      lea.l y1_k1_Y2(pc),a1
10:     sub.w y1,y2 ; y2 - y1 = dy
11:     bpl.s \d
12:     addq.l #1,a1
13:     neg.w dy
14:     \d
15:     lea.l DX_gr_DY(pc),a2
16:     cmp.w dy,dx ; dx > dy?
17:     bpl.s \d1 ; ja
18:     addq.l #1,a2
19:     exg dx,dy ; Algorithmus erwartet dx > dy
20:     \d1
21:     move.b (a0),d5 ; Schnittmenge der # Oktantenmengen
22:     and.b (a1),d5 ; bilden.
23:     and.b (a2),d5 ; Ergebnis : Gesuchter Oktant
24:     move.w #8,d4
25:     \l
26:     asr.b #1,d5 ; schiebt ins carry
27:     dbcs.s d4,\l ; nimmt Werte von 8 - 1 an
28:     move.w d4,d5 ; enthaelt nun Oktant
29:     \s2
30:     lea.l Oktanten(pc),a0 ; Oktant als Offset verdoppeln
31:     add.w d5,d5 ; == Tabelle wortweise angelegt
32:     move.w 0(a0,d5.w),d5 ; Oktant blittergerecht holen
33:     einschliesslich des Linemodus
34:     \Linie
35:     .....
36:     * Tabelle der Oktanten (1 - 8)
37:
38:     X1_k1_X2 dc.b %11000011 ; Okt1--> Okt8
39:     X2_k1_Y2 dc.b %00111100 ; entspr. %12345678
40:     Y1_k1_Y2 dc.b %11110000
41:     DX_gr_DY dc.b %00001111
42:     DX_gr_DY dc.b %10011001
43:     DX_gr_DY dc.b %01100110
44:

```

Listing 9: Die Oktanten nun als eine Menge von Bits. Der Codeteil kann entsprechende Teile in Listing 8 ersetzen, um == Ausfuehrungszeit zu gewinnen und das Programm zu kuerzen.

```

1:      sub.w x1,x2 ; ergibt delta x (dx)
2:      bml.s \Okt3456 ; x2/y2 in Oktant 3-6
3:      ; sonst 1,2,7 oder 8
4:      sub.w y1,y2 ; bilde dy
5:      bml.s \Okt78 ; x2/y2 in 7 oder 8
6:      cmp.w dy,dx
7:      bml.s \Okt2 ; dy > dx, folglich Oktant #
8:      moveq.l #OCTANT1+LINEMODE,d5
9:      bra.s \Linie
10:     \Okt2
11:     exg dx,dy ; Algorithmus erwartet dx groesser dy
12:     moveq.l #OCTANT2+LINEMODE,d5
13:     bra.s \Linie
14:     \Okt78
15:     neg.w dy ; Absolutwert dy bilden
16:     cmp.w dy,dx
17:     bml.s \Okt7
18:     moveq.l #OCTANT8+LINEMODE,d5
19:     bra.s \Linie
20:     \Okt7
21:     exg dx,dy ; Algorithmus erwartet dx groesser dy
22:     moveq.l #OCTANT7+LINEMODE,d5
23:     bra.s \Linie
24:     \Okt3456
25:     neg.w dx ; Absolutwert bilden
26:     sub.w y1,y2 ; ergibt dy
27:     bml.s \Okt56 ; sonst Oktant 3 oder #
28:     cmp.w dy,dx
29:     bml.s \Okt3
30:     moveq.l #OCTANT4+LINEMODE,d5
31:     bra.s \Linie
32:     \Okt3
33:     exg dx,dy ; Algorithmus erwartet dx groesser dy
34:     moveq.l #OCTANT3+LINEMODE,d5
35:     bra.s \Linie
36:     \Okt56
37:     neg.w dy ; Absolutwert dy bilden
38:     cmp.w dy,dx
39:     bml.s \Okt6
40:     moveq.l #OCTANT5+LINEMODE,d5
41:     bra.s \Linie
42:     \Okt6
43:     exg dx,dy ; Algorithmus erwartet dx groesser dy
44:     moveq.l #OCTANT6+LINEMODE,d5
45:

```

Listing 10: Alternativer Ansatz, den Oktanten fuer den Blitter zu bestimmen. Routine kann den Teil von "Ziehelinie" vor der Marke "\Linie" in Listing 8 ersetzen. Welche Loesung ist fuer Sie anschaulicher? Welche ist performanter?

Listings zum Assemblerkurs

## NEWLOTT GbR

### HARDWARE, SOFTWARE UND ZUBEHOER

GoethestraÙe 30 · 4100 Duisburg 18 (Walsum)

24-h-Bestellservice Telefon 02 03/49 57 97

Geschäftszeiten: Mo-Fr 16:00-18:30 Sa 10:00-14:00 Uhr

#### Leerdisketten

3,5" NoName 2DD 135 Tpi 50 Stück = 75,- 100 Stück = 149,-  
3,5" EDIXA 2DD 135 Tpi 50 Stück = 105,- 100 Stück = 199,-

#### Laufwerke fuer AMIGA 500/1000/2000

3,5" WINNER Drive Slimline (nur 25,4 mm Höhe) 225,-

5,25" WINNER Drive 40/80 Track umschaltbar 269,-

alle Laufwerke mit durchgefuehrtm Bus, abschaltbar

3,5" Laufwerk fuer A2000 intern im Einbausatz 165,-

#### Speichererweiterungen

0,5 MByte Ramkarte fuer A500 m. Uhr, abschaltbar 199,-

1 MByte Ramkarte Commodore A2058 fuer AMIGA 2000

bestückt mit 2 MByte Ram 1098,-

#### Harddisks

20 MByte Harddisk COMMODORE A590 fuer Amiga 500

SCSI-Controller, autobootend, int. auf 2 MByte RAM aufrüstbar

bestückt mit 0 MB Ram 978,- 0,5 MB Ram 1138,-

1,0 MB Ram 1278,- 2,0 MB Ram 1568,-

WINNER Filecard fuer AMIGA 2000 mit Autobootkarte

20 MB 879,- 31 MB 968,- 40 MB 1198,- 47 MB 1298,-

WINNER-Autoboot-Einbausatz fuer AMIGA 2000

30 MB 890,- 63 MB 1288,-

Alle Harddisks sind bereits unter FFS formatiert und mit

WB 1.3d installiert, nach Einbau sofort betriebsbereit.

FAT-AGNUS 8372A mit Einbauanleitung 159,-

Kickstartumschaltplatine mit ROM V1.2 oder 1.3 98,-

Bootselector DF0: bis DF2: 48,-

Weitere Harddisks, Drucker und Zubehoer auf Anfrage

#### PUBLIC DOMAIN (Stand 3/90)

Stückpreis 4,50 ab 10 = 3,50 ab 50 = 2,80 ab 100 = 2,50 DM

Fish -320, Kickstart -260, Taifun -120, Erotik -32, Auge -40

Franz -47, Cactus -35, Tornado -30, Nicolas -10

Lieferung per NN, Versandkosten # DM, Ausland Vorkasse

## 2400 bps MODEMS



#### TORNADO 2400E

Tischgerät incl. Steckernetzteil  
für alle Rechner mit RS232/V.24

348,-

#### MAXMODEM 2400 MNP5

Wie oben, mit MNP5 Übertragungs-  
protokoll bis 4800 bps eff. Geschw.

599,-

#### TECS 2400H, PC-Karte, halbe

Länge, COM1: bis COM4:  
konfigurierbar

298,-

Lieferung per Nachnahme, komplett mit engl. Handbuch und Telefonkabel (USA),  
1 Jahr Garantie, Rückgaberecht innerh. von 8 Tagen ohne Angabe von Gründen.

#### Leistungsmerkmale:

2400, 1200, 600, 300 bps CCITT V.22bis/V.22/V.21, 1200, 300 bps Bell 212a/103, kompatibel mit HAYES SMARTMODEM 2400 (AT-Kommandos), automatische Wahl (Ton oder Impuls), Autoanswer, Konfiguration speicherbar.  
Der Betrieb dieser Modems am öffentlichen Postnetz in der # und in West-Berlin # verboten und unter Strafe gestellt.

Carl Schewe (GmbH & Co.), Abt. Modems

Essener Str. 97, 2000 Hamburg 62

Tel. (0 40) 5 27 03 21, Fax (0 40) 5 27 66 54

# Amiga-BASIC

## Programmieren mit Erfolg

### Hallo, hallo – Amiga ruft BASIC

Im letzten Kursteil kommen noch einige Schmankerl auf Sie zu. Wir beschäftigen uns mit einem Grundprinzip modularer Programmierung, hören, wie unser Amiga mit uns spricht, und zaubern nebenbei noch Töne aus dem Lautsprecher heraus.

**E**in Ziel sollte jeder vor Augen haben, der sich etwas intensiver mit Programmiersprachen, genauer gesagt mit der Entwicklung von Programmen, beschäftigt: Programme modular zu entwerfen, daß heißt, das Programm in möglichst kleine, eigenständige Programmteile aufzuspalten, die so jederzeit einzeln getestet werden können.

### Hallo, Hauptprogramm ruft Unterprogramm

Ein weiterer Punkt ist, Programmteile, die ständig wieder gebraucht werden, zu einer Routine zusammenzufassen und als Unterprogramm laufen zu lassen. Der Vorteil solcher Routinen liegt klar auf der Hand: Sie sparen eine Menge an Zeit, Aufwand und auch Speicherplatz. Die Programme werden dadurch wesentlich übersichtlicher, besonders in Bezug auf die Fehlerkontrolle. Wie sieht das Ganze nun programmiertechnisch aus?

Unterprogramme sind in sich geschlossene Programmteile oder auch logisch zusammengehörige Einheiten. Solche Einheiten lassen sich auf verschiedene Art und Weise definieren: Die kürzeste Form eines Unterprogramms ist die Funktion. In einer Funktion können beispielsweise komplizierte Formeln, deren Ergebnis als konstanter Wert im Programm eingebunden werden soll, definiert und berechnet werden. Jedoch ist zu beachten, daß die Funktionsdefinition nicht mehr als 80 Zeichen (entspricht einer Programmzeile in BASIC) umfassen darf. Die Syntax dieses Befehls lautet:

```
DEF FNName[(Arg[,Arg]...)]
= Funktionsdefinition
```

Kursfahrplan
Teil 1: BASIC zum Kennenlernen
Teil 2: Dem Spaghetti-Code keine Chance
Teil 3: Vom Strich zum Bild
Teil 4: Dateien über Dateien
Teil 5: Unter dem Programm ein Unterprogramm

Kommen wir zu den einzelnen Parametern: "Name" ist der Name der Variablen, der als Funktionsname dient und ohne Leerstelle hinter FN stehen muß. "Arg" ist der Variablenname, der beim Aufruf der Funktion durch die aktuelle Variable bzw. den aktuellen Wert ersetzt wird. Der Parameter "Funktionsdefinition" beinhaltet eine Rechenvorschrift oder Rechenformel, nach der der Funktionswert berechnet werden soll. Es kann jeweils nur eine Rechenvorschrift in einer Funktionsdefinition angegeben werden. Die Funktionsdefinition kann ein numerischer oder ein Zeichenkettenausdruck sein. Dabei ist jedoch darauf zu achten, daß dieser Ausdruck vom Typ her mit dem Typ von "Name" übereinstimmt.

Ein kurzes Beispiel soll die Arbeitsweise des Befehls DEF FN verdeutlichen. Begeben wir uns in die mathematischen Gefilde und erinnern uns an unseren Mathelehrer, der uns immer wieder mit der Frage nach dem Satz des Pythagoras gelöchert hat. Na, wissen Sie ihn noch? Richtig, im rechtwinkligen Dreieck ist das Quadrat über der Hypotenuse so groß wie die Summe der Quadrate über den Katheten. Demzufolge können wir nun eine Funktion entwerfen, die die Länge der Hypotenuse berechnen soll. Ausgehend von unserer Pythagoras-Formel ( $a^2 + b^2 = c^2$ )

gehen wir in unseren BASIC-Interpreter und geben als Definition folgendes ein:

```
DEF FNC(a,b)=SQR((a*a)+(b*b))
```

Möchte man nun verschiedene Eingaben machen, geben wir noch folgendes ein:

```
INPUT "Geben Sie Werte ein";a,b
PRINT "Die Länge der Hypothenu-
    betr gt: "FNC(a,b)
```

Damit haben wir ein kurzes Programmbeispiel erstellt, mit dem sich bereits die verschiedenen L ngen unserer Hypothenusen berechnen lassen. Ein weiteres kleines Beispiel zeigt, wie in einer Endlosschleife viele unterschiedliche Rechtecke per Rundungsfaktor unter Verwendung der obengenannten Funktion (DEF FNC...) erzeugt werden. Gehen Sie wieder in das LIST-Fenster Ihres BASIC-Interpreters, und geben Sie folgendes Beispiel ein:

```
start:
DEF FNC=SQR((a*a)+(b*b))
a=INT(RND*100)+10
b=INT(RND*100)+10
AREA (0,0)
AREA (INT(FNC),0)
AREA (INT(FNC),INT(FNC))
AREA (0,INT(FNC))
AREAFILL x
IF x=0 THEN x=1 ELSE x=0
GOTO start
```

Da dies wie gesagt eine Endlosschleife ist, k nnen Sie dieses Beispiel in der Men leiste Ih-

res BASIC-Interpreters mit STOP im RUN-Men  beenden. Falls Sie jedoch ein wenig herumprobieren m chten,  ndern Sie dieses Miniprogrammchen doch einfach so ab, da  Sie beispielsweise ein Ende-Kriterium hinzuf gen oder  hnliches. Da diese Funktionen jedoch auf eine Programmzeile beschr nkt sind, wenden wir uns einer gel ufigeren und komfortableren M glichkeit, Unterprogramme zu erstellen, zu: der GOSUB-label-Konstruktion.

### Spr nge hin und her im Programm

Mit diesem Konstrukt kann von einem Hauptprogramm aus in Subroutinen verzweigt und auch wieder zur ck ins Hauptprogramm gesprungen werden, wenn der Interpreter auf ein RETURN trifft, das die Subroutine beendet. Die Anzahl der Subroutinen, die in einem Programm verwendet werden, ist unbegrenzt, jedoch kennen diese Routinen keine lokalen Variablen. Das hei t, alle Variablen, die in einer Subroutine enthalten sind, sind auch im Hauptprogramm bekannt. Hier liegt aber auch "der Hase im Pfeffer": Bei der Definition der einzelnen Variablen mu  man unbedingt daf r sorgen, da  diese Variablen nicht bereits im Hauptprogramm verwendet wurden, da dies sehr fatale Folgen h tte. Wird eine Variable, die im Hauptprogramm definiert ist, auch f r die Subroutine neu definiert,  ndert sie somit ihren Wert und demzufolge auch den Wert, den sie im Hauptprogramm hat.

Damit ist gleichzeitig die Funktionsweise solcher Subroutinen stark eingeschr nkt, da sie nicht als eigenst ndiges Programm lauff hig sind, sondern immer nur im Zusammenhang und in Verbindung mit dem Hauptprogramm laufen k nnen. Auch die Fehlersuche ist ein kritischer Faktor, den man bei der Verwendung von Subroutinen in eigenen Programmen bedenken sollte. Innerhalb von Subroutinen k nnen jeder Zeit noch weitere Subroutinen aufgerufen werden. Hier setzt lediglich der Hauptspeicherplatz diverse Grenzen. Nach der RETURN-Anweisung kehrt die Subroutine wieder zum Hauptprogramm zur ck und setzt direkt mit der n chsten Anweisung, die der GOSUB label-Anweisung folgt, fort. Die An-



wendung solcher Anweisungen ist mit einer gewissen Vorsicht zu genießen, da die übrigen GOSUB-, CALL-, WHILE- und FOR-Anweisungen, die zur Zeit des Aufrufs der Subroutine aktiv waren, auch hinterher aktiv bleiben. Das heißt, sie belegen somit Bezugsadressen auf dem Stapelspeicher. Die Syntax der Anweisung GOSUB label sieht folgendermaßen aus:

```
GOSUB label
...
...
RETURN
```

Listing 1 zeigt in einem einfachen Beispiel die Funktionsweise dieser Anweisung.

## Hallo, hallo, rufe Unterprogramm

Wenden wir uns nun den "richtigen" Unterprogrammen zu. Richtig insofern, da diese Unterprogramme auch tatsächlich in sich geschlossene Einheiten bilden. Es können in der Tat lokale (im wahrsten Sinne des Wortes) Variablen verwendet werden, die **nur** dem Unterprogramm, jedoch nicht dem Hauptprogramm bekannt sind. Dies führt natürlich auch dazu, daß Variablen, die im Hauptprogramm verwendet werden, auch im Unterprogramm verwendet werden können, ohne sich gegenseitig "ins Gehege zu kommen". Diese Unterprogramme können, da sie ja eigenständigen Charakter besitzen, alleine stehen und somit auch einzeln getestet werden. Die Fehlersuche läßt sich so gut eingrenzen. Ein weiterer Vorteil der sogenannten Unterprogrammtechnik liegt darin begründet, daß hier wirklich modulares Programmieren möglich ist. Das heißt, man kann sich selbst seine eigene Bibliothek zusammenstellen, indem man häufig wiederkehrende Routinen zu Unterprogrammen zusammenfaßt.

Schauen wir uns zunächst die Syntax dieser Unterprogrammtechnik an. Wir sprechen im folgenden nicht mehr vom Unterprogramm, sondern verwenden den passenderen Ausdruck SUB-Programm. Der Aufruf eines SUB-Programms erfolgt mit

CALL Name (Parameterliste)

wobei "Name" der Name des aufzurufenden SUB-Programms ist und "Parameterliste" die zu übergebenden Variablen bezeichnet. Die Syntax

```
der eigentlichen Unterprogramm-
anweisung lautet dann:
SUB Name (Parameterliste)
STATIC
END SUB
EXIT SUB
```

Gehen wir näher auf die einzelnen Parameter ein: Mit "Name" ist die Bezeichnung des jeweiligen SUB-Programm-Aufrufs gemeint. "Name" kann bis zu 40 Zeichen umfassen, darf jedoch in keiner anderen SUB-Anweisung verwendet werden. In "Parameterliste" können einfache oder Feldvariablen (Stichwort: Dimensionsierung) angegeben werden, die Anzahl ist lediglich durch die Länge einer logischen Programmzeile (254 Zeichen) begrenzt. Die einzelnen Parameter werden durch Kommata getrennt. Bei Feldvariablen ist darauf zu achten, daß diese in runde Klammern "()" gesetzt werden. Ein wesentlicher Aspekt kommt dem Begriff "STATIC" zu. ("STATIC" heißt statisch oder unverändert.) Dieser besagt, daß zwischen zwei SUB-Programmaufrufen alle Variablen ihre Werte behalten. Ein weiterer wichtiger Punkt ist, daß alle Variablen vom Hauptprogramm **nicht** verändert werden können. Zudem setzt "STATIC" fest, daß das SUB-Programm nicht rekursiv ist, das heißt, sich selbst und andere SUB-Programme aufrufen kann. "END SUB" bedeutet, daß das SUB-Programm beendet ist und das die Programmsteuerung an das Hauptprogramm zurückgegeben wird. Dabei wird die dem SUB-Programmaufruf folgende Anweisung im Hauptprogramm ausgeführt. Mit der "EXIT SUB"-Anweisung ist es möglich, aus dem SUB-Programm "auszusteigen". Diese Art Ausprung ist beispielsweise bei bedingten Abfragen angeraten. Wir wollen aber nicht näher darauf eingehen.

Ein kleines Beispiel soll zeigen, ob Variablen in einer "SUB STATIC"-Anweisung verändert werden oder nicht. Gehen Sie in Ihren BASIC-Interpreter, und geben Sie im LIST-Fenster ein:

```
a=1
b=2
c=3
CALL Unter
PRINT a,b,c
FOR i=1 TO 5000:NEXT
END
SUB Unter STATIC
a=5:b=6:c=7
END SUB
```

Sie werden am Ergebnis feststellen, ob und wo sich etwas ändert. Zum Gegensatz tippen Sie bitte folgendes in Ihren BASIC-Interpreter ein:

```
a=1
b=2
c=3
CALL Unter
PRINT a,b,c
FOR i=1 TO 5000:NEXT
END
SUB Unter STATIC
SHARED c
a=5:b=6:c=7
END SUB
```

Sie werden sicherlich bemerkt haben, daß hier ein deutlicher Unterschied besteht. Das erste Beispiel zeigt, daß Variablen, die im Hauptprogramm definiert wurden, nicht durch das SUB-Programm verändert werden, da diese Variablen **lokale** Variablen sind und somit nur für ihren speziellen Bereich gelten, wo Sie definiert wurden. Demgegenüber zeigt das zweite Beispiel, wie mit Hilfe der "SHARED"-Anweisung eine Variable einen **globalen** Charakter erhält. Das heißt, diese Variable ist gleichzeitig im Hauptprogramm und im SUB-Programm gültig.

Ein wichtiger Hinweis: Folgende Anweisungen dürfen in SUB-Programmen **auf gar keinen Fall** verwendet werden:

- CLEAR
  - COMMON
  - DECLARE FUNCTION
  - DEF FN
  - SUB STATIC
  - LIBRARY (sollte aus Zeitgründen nicht in einem SUB-Programm verwendet werden).
- Damit wollen wir es beim Thema "Unterprogramme" bewenden lassen und uns einigen akustischen Bonbons zuwenden.

Sie haben sich Ihren Amiga bestimmt nicht nur gekauft, um damit damit zu spielen. Sicherlich waren Sie genauso neugierig wie ich, was man mit der "Spielekiste" noch so alles anstellen kann. Eine große Stärke haben wir ja bereits kennengelernt: die grafischen Fähigkeiten des Amiga. Damit nicht genug, nein, unser Amiga kann sich richtiggehend mit uns unterhalten. Und, welcher "stinknormale" PC kann das schon?

## Hilfe, mein Computer spricht mit mir!

Praktisch ist die ganze Sache auch noch, denn in Amiga-BASIC ist die Sprachausgabe

fast ein Kinderspiel. Mit dem "SAY"-Befehl kann Sprache in fast jeder beliebigen Landessprache ausgedrückt werden, vorausgesetzt man ist der phonetischen Lautschrift des entsprechenden Landes kundig. Die Syntax dieses Befehls lautet:

SAY Zeichenkette [,Modus]

Sie brauchen aber keine Angst zu haben, so schlimm, wie es gerade klingt, ist es nicht. Um etwas mit "SAY" sagen zu können, nimmt man sich zunächst eine "Zeichenkette" vor, die aus Phonem-Codes (Phonem entspricht dem deutschen "Laut"; es handelt sich also um Laut-Codes) zusammengesetzt ist. Diese ergeben dann den Text, den man hören möchte. Ist ja alles schön und gut, doch wo bekomme ich diese Phonem-Codes denn her, werden Sie fragen.

Diese Frage hat sicherlich ihre Berechtigung. Es gibt hier mehrere Möglichkeiten, die man in Betracht ziehen kann. Zum einen schlägt man in seinem Amiga-Basic-Handbuch nach. Dort ist der Sprachausgabe ein ganzes Kapitel gewidmet. Eine weitere Möglichkeit ist die Verwendung eines Wörterbuchs. Dabei ist es egal, ob man ein deutsches oder ein englisches Wörterbuch zu Rate zieht. Wesentlich ist nur, daß hinter jedem Wort in Klammern der phonetische Ausdruck steht. Diesen kann man beruhigt hinter dem SAY-Befehl eingeben. Die letzte Möglichkeit besteht darin, die Worte, die man per SAY-Befehl ausgeben will, so zu schreiben, wie man sie im wahrsten Sinne des Wortes auch spricht.

Kommen wir nun zum Parameter "Modus": Dieser Parameter entspricht einem Ganzzahlenfeld mit mindestens neun Elementen (es wird dabei von 0 bis einschließlich 8 gezählt). Beschäftigen wir uns kurz mit den einzelnen Elementen.

Element 0: Grundfrequenz in Hertz, dabei können Werte zwischen 65 und 320 angegeben werden.

Element 1: Betrifft die Modulation, hier können zwei Werte angegeben werden. 0 für Silbennmodulation und Betonung und 1 für eine monotone, roboterhafte Sprache.

Element 2: Regelt die Sprechgeschwindigkeit, wobei Werte zwischen 40 und 400 angegeben werden können.

## BASIC-Glossar (Teil 4)

In unserem BASIC-Glossar finden Sie außer den Befehlen, die im Text beschrieben werden, auch artverwandte Befehle:

**[CALL]Sprungmarke[(Argumentliste)]**

oder

**[CALL]num Var [(Argumentliste)]**

Mit dieser Anweisung wird ein AmigaBASIC-Unterprogramm (mit SUB... definiert, siehe obere Syntax), ein Maschinenprogramm (siehe untere Syntax) oder eine Maschinensprache-Subroutine aus einer Bibliothek (mit der LIBRARY-Anweisung definiert, siehe untere Syntax) aufgerufen. Das Schlüsselwort CALL muß nicht unbedingt verwendet werden. Einige Beispiele:

■ Aufruf von AmigaBASIC-Unterprogramm:

SUB Unterprogramm(a,b) STATIC

END SUB

CALL Unterprogramm(a,b)

Hier können Variablen oder Ausdrücke übergeben werden, wobei Variablen durch Referenz und Ausdrücke durch ihren Wert übergeben werden.

■ Aufruf von Maschinenprogramm:

CALL Routine(VarPTR(x))

Dies ist die einzige Möglichkeit, die Programmsteuerung an ein externes Unterprogramm zu übertragen. Wichtig ist, daß die aufgerufene Maschinenroutine auf einer geraden Speicherstelle beginnt, da sich der Rechner sonst ins Nirwana verabschiedet.

■ Aufruf von Bibliotheksroutinen:

LIBRARY "graphics.library"

CALL Draw(30,40)

Diese Bibliotheksroutinen sind spezielle Amiga-Programmdateien, die dynamisch, das heißt, während der Laufzeit, in AmigaBASIC eingebunden werden.

**CLEAR[, [Prg][, Stap]]**

Dieser Befehl dient dazu, die Inhalte aller Variablen entweder durch Setzen auf 0 oder durch Setzen eines Leerstrings zu löschen.

**COMMON Variable[, Variable]...**

Die bei dieser Anweisung angegebenen Variablen werden an ein Programm übergeben, das mit der CHAIN-Anweisung aufgerufen wurde. Werden Feldvariablen übergeben, so werden diese durch Klammern () hinter dem jeweiligen Namen ergänzt.

**DECLARE FUNCTION Name[(Parameterliste)] LIBRARY**

Mit dieser Anweisung wird eine Maschinensprache-Routine aus einer Bibliothek, die mit LIBRARY angegeben wird, als eine aufrufbare Funktion deklariert.

**DEF FNName[(Arg[, Arg]...)] = Funktionsdefinition**

Der Anwender kann eine eigenständige Funktion definieren und benennen.

**END SUB**

Dies bezeichnet das Ende eines Unterprogramms. Die Programmsteuerung wird an die Anweisung übergeben, die dem Unterprogrammaufruf im Hauptprogramm folgt.

**EXIT SUB**

Mit dieser Anweisung kann an jeder beliebigen Stelle das Unterprogramm verlassen werden. Auch hier geht die Programmsteuerung an die dem Unterprogrammaufruf folgende Anweisung im Hauptprogramm weiter.

**v = FRE(x)**

Diese Funktion liefert je nach dem Wert von x Aussagen über freien Speicherplatz in Bytes: Wenn x = -1 liefert die Aussage über den noch verfügbaren Speicherplatz. Wenn x = -2 liefert die Aussage über den noch verfügbaren AmigaBASIC-Stapelspeicher.

Wenn x > -1 liefert die Aussage über den noch verfügbaren Speicher im AmigaBASIC-Datensegment.

**GOSUB Label**

..

**RETURN [Weiter]**

Mit dieser Anweisung erfolgt ein Sprung in eine Subroutine und die Rückkehr in das Hauptprogramm. Label ist der Name der betreffenden Subroutine. Weiter ist optional.

**LIBRARY "Dateiname"**

**LIBRARY CLOSE**

Mit dieser Anweisung wird eine Bibliothek mit Maschinensprache-Unterprogrammen für den Zugriff aus AmigaBASIC heraus geöffnet und mit LIBRARY CLOSE alle geöffneten Bibliotheken geschlossen.

**SAY Zeichenkette[, Modus]**

Gibt eine Zeichenkette von Laut-Codes in einer relativ verständlichen Sprache wieder, wobei mit dem Parameter "Modus" 9 verschiedene Elemente angegeben werden können, die die Ausgabe steuern.

**SHARED Variable[, Variable]...**

Mit dieser Anweisung können Variablen innerhalb eines mit SUB aufgerufenen Unterprogramms als global, das heißt, allgemein gültig, definiert werden.

**SOUND Frequenz, Dauer[, [Laut][, Kanal]]**

Mit dieser Anweisung wird ein Ton in einer Warteschlange abgelegt, die anschließend über einen entsprechend wählbaren Ton-Kanal abgespielt wird. Dabei sind jeweils Frequenz, Dauer und Lautstärke dieses Tons frei wählbar.

**SOUND WAIT**

Diese Anweisung legt die Ergebnisse aller folgenden SOUND-Anweisungen in eine Warteschlange.

**SOUND RESUME**

Diese Anweisung beendet den Wartezustand der SOUND WAIT-Anweisung.

**SUB Name[(Parameterliste)] STATIC**

Dies Anweisung leitet ein AmigaBASIC-Unterprogramm ein. STATIC legt dabei fest, daß die im Unterprogramm verwendeten Variablen lokal sind.

**v\$ = TRANSLATE\$(text)**

Mit dieser Funktion werden Folgen von Laut-Codes erzeugt und an die SAY-Anweisung übergeben. Text ist dabei eine Zeichenkette aus der angloamerikanischen Sprache.

**WAVE Kanal, Definition**

Mit dieser Anweisung lassen sich für einen anzugebenden Tonkanal (Kanal) Formen von Tonwellen definieren.

Element 3: Hier kann das Geschlecht des Sprechers eingestellt werden. 0 entspricht einer männlichen und 1 einer weiblichen Stimme.

Element 4: Die Stimmlage kann verändert werden von 5000 (tief) bis 28000 (hoch).

Element 5: Damit läßt sich die Lautstärke regeln - von 0 (kein Ton) bis 65 (größte Lautstärke).

Element 6: Hier wird der Tonkanal für die Sprachausgabe eingestellt.

Element 7: Einstellung des Synchronisationsmodus - 0 entspricht synchroner, 1 asynchroner Sprachausgabe.

Element 8: Dient zur Steuerung der Bearbeitung aufeinanderfolgender SAY-Anweisungen bei asynchroner

Sprachausgabe. Es sind drei Werte möglich: 0 entspricht der normalen Verarbeitung, bei 1 wird die Sprachausgabe abgebrochen, und bei 2 wird die Bearbeitung der aktuellen SAY-Anweisung abgebrochen.

Damit die SAY-Anweisung nicht so ganz alleine im Raum steht, haben sich die Pro-

grammierer von Amiga-BASIC eine spezielle Funktion geschaffen: TRANSLATE.

Diese Funktion liefert einen String, der die dem Text entsprechenden Phonem-Codes enthält. Genauer gesagt, TRANSLATE spricht den Text aus, den man SAY übergibt. Die Syntax dieses Befehls sieht wie folgt aus:



v\$=TRANSLATE\$(Text)

Sehen wir uns die Syntax etwas näher an. "v\$" ist die Variable, die der SAY-Anweisung übergeben wird. In dem Parameter "Text" ist ein Zeichenkettenausdruck, dessen Inhalt in der bisher vorliegenden Form in angloamerikanischer Sprache angegeben werden muß und der anschließend mit der SAY-Anweisung ausgegeben wird.

Ein kleines Beispiel:

```
a$=TRANSLATE$("fishers frits
fisht frisha fisha")
SAY a$
b$=TRANSLATE$("frisha fisha
fisht fishers frits")
SAY b$
```

Das Ergebnis dürfte Ihnen sicherlich irgendwo schon einmal begegnet sein, zumindest verdeutlicht es Ihnen die Wirkungsweise der TRANSLATE-Funktion. Ein kleines Programm (Listing 2) zeigt Ihnen eine Möglichkeit, mit der Sprache in Amiga-BASIC rea-

liert werden kann. Mit einiger Übung läßt sich Ihr Amiga bestimmt zu einem wahren Sprachkünstler entwickeln.

## Amiga - die Soundmaschine

Musik auf dem Amiga, wenn klingen da nicht die Ohren. Mit Amiga-BASIC ist Musik sogar mit einem einzigen Befehl möglich: SOUND heißt das Zauberwort, mit dem sich sogar kleine Lieder spielen lassen. Wie funktioniert dieser SOUND-Befehl? Es wird ein Ton mit frei wählbarer Frequenz, Dauer und Lautstärke in einer sogenannten Ton-Warteschlange abgelegt, die anschließend über einen gleichfalls wählbaren Ton-Kanal abgespielt wird. Schauen wir uns die Syntax an:

```
SOUND Frequenz,Dauer,[Laut][,Kanal]
SOUND WAIT
SOUND RESUME
```

Die einzelnen Parameter haben folgende Bedeutung: Bei

"Frequenz" kann ein Ganzzahlwert angegeben werden, der im Bereich zwischen 20 und 15000 liegen muß. Diese Werte entsprechen den Angaben in Hertz. Ein Beispiel: Ein Wert von 100 erzeugt unter Amiga-BASIC einen Ton von 100 Hertz. Die Tondauer wird mit dem Parameter "Dauer" geregelt, wobei hier ein numerischer Ausdruck im Bereich zwischen 0 und 77 angegeben werden muß. Mit "Laut" kann die Lautstärke des angegebenen Tons geregelt werden. Der Wert kann im Bereich von 0 (kleinste Lautstärke) bis 255 (größte Lautstärke) liegen. Mit "Kanal" können die Tonkanäle bestimmt werden, hier sind Werte zwischen 0 und 3 möglich.

Kommen wir zum Begriff "SOUND WAIT": Mit dieser Anweisung werden die Ergebnisse der nachfolgenden SOUND-Anweisungen in einer Warteschlange abgelegt, und zwar so lange, bis eine SOUND-RESUME-Anweisung

erfolgt. Damit kann eine Synchronisation der Töne aus den unterschiedlichen Tonkanälen des Amiga erfolgen.

Um unseren Amiga-BASIC-Kurs noch ein wenig abzurunden, stellen wir Ihnen zum Abschluß ein Programm vor, mit dem sich Adressen verwalten lassen. Sie können den Quelltext dem Listing 3 entnehmen. Bei diesem Programm sind viele der im gesamten Kurs erlernten und angewandten Befehle und Anweisungen enthalten.

Damit ist unser Kurs "Amiga-BASIC - Programmieren mit Erfolg" an seinem Ende angelangt. Keine Angst!!! Das Kapitel Amiga-BASIC ist damit längst noch nicht abgeschlossen. Wir greifen uns in den nächsten Ausgaben immer wieder kleinere Schmankerl heraus und zeigen Ihnen, wie Sie damit umgehen können. Amiga-BASIC läßt Sie also auch in Zukunft grüßen. Bis demnächst und gut "Üb".

(vb)

## Listings

```
000:718 REM *****
001:412 REM * unterprogramm.bas *
002:762 REM * Dies ist ein Beispiel fuer den Aufruf von *
003:746 REM * von Unterprogrammen mit Hilfe der GOSUB- *
004:474 REM * Anweisung. *
005:612 REM * Sprache: AmigaBASIC *
006:718 REM *****
007:000
008:000
009:008 PRINT "Hallo Hauptprogramm ruft Unterprogramm"
010:952 FOR i=1 TO 10000:NEXT
011:633 GOSUB unter1
012:393 LOCATE 20,1
013:228 PRINT "Das Hauptprogramm meldet sich wieder zurueck"
014:952 FOR i=1 TO 10000:NEXT
015:503 CLS
016:822 LOCATE 10,30
017:932 PRINT "Tschuess"
018:500 END
019:316 unter1:
020:503 CLS
021:436 LOCATE 8,20
022:216 PRINT "Hallo, hier ist das erste Unterprogramm"
023:820 LOCATE 10,20
024:138 PRINT "weiter mit Taste"
025:444 WHILE INKEY$="" :WEND
026:634 GOSUB unter2
027:982 RETURN
028:000
029:318 unter2:
030:503 CLS
031:836 LOCATE 12,20
032:408 PRINT "Juhu, hier meldet sich Unterprogramm zwei"
033:852 LOCATE 14,20
034:138 PRINT "weiter mit Taste"
035:444 WHILE INKEY$="" :WEND
036:982 RETURN
037:026
```

Listing 1: Die Unterprogrammtechnik dargestellt am Beispiel einer GOSUB-Anweisung

```
000:718 REM *****
001:940 REM * sprache.bas *
002:584 REM * Dieses Programm verdeutlicht die Sprachaus- *
003:840 REM * gabe mit Hilfe der SAY-Anweisung *
004:612 REM * Sprache: AmigaBASIC *
005:718 REM *****
006:000
007:727 WINDOW 1:CLS
008:074 start:
009:168 CLEAR
010:188 PRINT "Was soll ich sagen? Gib einen Satz ein!"
011:468 PRINT
012:112 INPUT Satz$
013:686 FOR n=0 TO 1
014:829 READ Feld%(n)
015:160 NEXT
016:273 Text$=TRANSLATE$(Satz$)
```

```
017:468 PRINT
018:948 PRINT Text$,Feld%
019:933 SAY Text$,Feld%
020:124 PRINT "Nochmal ?? {j/n}"
021:076 WHILE w$="" :w$=INKEY$:WEND
022:380 IF UCASE$(w$)="J" THEN GOTO start
023:000
024:660 Feldwerte:
025:348 DATA 140,0,220,1,28000,64,11,1,0
026:026
```

Listing 2: So kann beispielsweise die Sprachausgabe in Amiga-BASIC aussehen

```
000:518 REM *****
001:892 REM * adressen.bas *
002:832 REM * Adressverwaltungsprogramm *
003:396 REM * (c) 1990 Steffen Kaiser & AMIGADOS *
004:612 REM * Sprache: Amigabasic *
005:518 REM *****
006:000
007:000
008:000
009:345 DIM nr(200)
010:000
011:201 GOSUB oeffne
012:164 GOSUB Fenster
013:920 GOSUB Farben
014:815 GOSUB Gadgets
015:055 GOSUB Maske
016:656 Marke:
017:983 GOSUB choose
018:961 ON ch GOSUB rueck,vor,change,watch,neu,Talk,Dru,loes
ch,raus
019:491 GOTO Marke
020:500 END
021:000
022:000
023:452 oeffne:
024:800 OPEN "R",#1,"df0:telefon.dat",104
025:232 FIELD #1,20 AS n$,20 AS v$,20 AS s$,4 AS p$,20 AS o$,
20 AS t$
026:408 Anzahl = LOF(1)/104
027:982 RETURN
028:000
029:000
030:370 Fenster:
031:302 SCREEN 1,640,256,3,2
032:905 WINDOW 1,"Adressverwaltung",(3,0)-(626,222),20,1
033:549 COLOR 5
```

```

034:971 PAINT (1,1)
035:316 COLOR 1,0
036:982 RETURN
037:000
038:890 Farben:
039:176 PALETTE 0,0,0,0
040:281 PALETTE 1,1,1,1
041:321 PALETTE 2,1,0,1
042:372 PALETTE 3,0,1,0
043:433 PALETTE 4,0,0,1
044:297 PALETTE 5,1,5,5
045:565 PALETTE 6,0,1,1
046:759 PALETTE 7,5,1,3
047:982 RETURN
048:000
049:000
050:672 Gadgets:
051:039
052:526      Hier wird der Balken mit den Gadgets gezeichnet
053:039
054:194 LINE (1,154)-(629,155),6,bf
055:394 LINE (1,180)-(629,181),6,bf
056:526 FOR i=0 TO 8
057:507   x1=i*77+1
058:642   x2=i*77+8
059:642 LINE(x1,155)-(x2,180),6,bf
060:818 LINE (x2+3,157)-(x2+67,178),0,bf
061:393 NEXT i
062:000
063:000
064:543 REM
065:798 REM      Hier werden die Gadgets gezeichnet
066:543 REM
067:543 REM
068:458 REM      Exit - Gadget
069:543 REM
070:018 LINE (570,161)-(590,174),7,bf
071:058 LINE (585,165)-(588,170),4,bf
072:956 LINE (581,167)-(587,166),0
073:604 LINE (587,168)-(587,169),0
074:543 REM
075:666 REM      Disketten - Gadget
076:543 REM
077:205 LINE (492,162)-(517,173),2,bf :REM Diskette
078:158 LINE (513,163)-(515,164),1,bf :REM Schreibschutz
079:456 LINE (498,168)-(512,173),6,bf :REM Metall
080:460 LINE (494,162)-(513,166),4,bf :REM Aufkleber
081:000
082:543 REM
083:930 REM      Drucker - Gadget
084:543 REM
085:307 LINE (412,170)-(442,173),3,bf :REM Drucker(1)
086:069 LINE (412,168)-(442,170),6,bf :REM (2)
087:223 LINE (414,166)-(440,168),2,bf :REM (3)
088:496 LINE (416,162)-(438,166),1,bf :REM Papier
089:067 LINE (410,168)-(412,170),1,bf :REM Rolle 1
090:268 LINE (442,168)-(444,170),1,bf :REM Rolle 2
091:695 PSET (418,163),0:PSET (436,163),0 :REM
092:136 PSET (418,165),0:PSET (436,165),0 :REM Perforation
093:543 REM
094:994 REM      Kopf - Gadget
095:543 REM
096:000
097:066 LINE(339,162)-(363,174),2,bf :REM kopf
098:407 LINE(343,164)-(347,165),6,bf :REM auge
099:807 LINE(355,164)-(359,165),6,bf :REM auge
100:023 LINE(351,165)-(351,166),3 :REM nase
101:082 LINE(350,167)-(352,168),3,bf
102:604 LINE(343,170)-(359,171),0,bf :REM mund
103:821 LINE(343,172)-(359,172),1
104:241 FOR i=0 TO 10 STEP 5
105:893   LINE(345+i,170)-(347+i,170),1
106:741   PSET(346+i,171),1
107:444   LINE(345+i,172)-(347+i,172),0
108:160 NEXT
109:543 REM
110:466 REM      Eingabe - Gadget
111:543 REM
112:000
113:794 LINE (259,163)-(271,168),3,bf
114:146 LINE (271,164)-(279,167),2,bf
115:842 LINE (275,167)-(278,172),2,bf
116:602 LINE (262,174)-(267,176),1,bf
117:010 LINE (269,173)-(274,175),1,bf
118:810 LINE (276,172)-(281,174),1,bf
119:802 LINE (283,171)-(288,173),1,bf
120:543 REM
121:594 REM      Sicht - Gadget
122:543 REM
123:000
124:784 LINE (178,161)-(214,174),2,bf
125:301 FOR b=0 TO 7
126:230   LINE (182,163+b)-(210,163+b),b
127:386 NEXT b
128:385 FOR b= 15 TO 40 STEP 5
129:618   LINE (168+b,172)-(170+b,173),0,bf
130:386 NEXT b
131:543 REM
132:514 REM      Aenderungs - Gadget
133:543 REM
134:000
135:245 LINE (94,174)-(107,174),1
136:861 LINE      -(110,172),1
137:133 LINE      -(112,174),1
138:901 LINE      -(142,174),1

```

```

139:298 LINE (106,163)-(114,170),2,bf
140:002 LINE (115,165)-(137,167),6,bf
141:543 REM
142:722 REM      Vor-Zurueck - Gadget
143:543 REM
144:433 a=3
145:445 LINE (31,162)-(16,167),1
146:861 LINE      -(31,173),1
147:741 LINE (36,162)-(21,167),1
148:501 LINE      -(36,173),1
149:205 LINE (50,162)-(65,167),1
150:245 LINE      -(50,173),1
151:501 LINE (55,162)-(70,167),1
152:759 LINE      -(55,173),1 :REM      <==
153:698 LINE (41,155)-(45,179),6,bf
154:100 LINE (43,155)-(43,179),0
155:039
156:868      Command-Text-Rahmen
157:039
158:834 LINE (200,194)-(430,212),6,bf
159:682 LINE (205,196)-(425,210),0,bf
160:000
161:982 RETURN
162:000
163:000
164:000
165:039
166:703      Erstellen der Bildschirmmaske
167:039
168:664 Maske:
169:345 COLOR 7,5
170:000
171:467 LOCATE 26,61 :PRINT Anzahl" Datensätze";
172:946 LOCATE 3,10 :PRINT "      Name : "
173:838 LOCATE 3,50 :PRINT "Kontakt:"
174:506 LOCATE 5,10 :PRINT "      Vorname : "
175:266 LOCATE 8,10 :PRINT "      Strasse : "
176:706 LOCATE 10,10:PRINT "Postleitzahl : "
177:826 LOCATE 12,10:PRINT "      Ort : "
178:562 LOCATE 15,10 :PRINT "      Telefon : "
179:000
180:000
181:114 LINE (193,13)-(366,25),0,bf
182:514 LINE (195,14)-(364,24),2,b
183:000
184:258 LINE (193,29)-(366,41),0,bf
185:514 LINE (195,30)-(364,40),2,b
186:000
187:770 LINE (193,53)-(366,65),0,bf
188:842 LINE (195,54)-(364,64),2,b
189:000
190:330 LINE (193,69)-(239,81),0,bf
191:050 LINE (195,70)-(237,80),2,b
192:000
193:426 LINE (193,85)-(366,97),0,bf
194:170 LINE (195,86)-(364,96),2,b
195:000
196:610 LINE (193,109)-(366,121),0,bf
197:922 LINE (195,110)-(364,120),2,b
198:000
199:378 LINE (450,29)-(600,121),0,bf
200:306 LINE (452,30)-(598,120),2,b
201:000
202:982 RETURN
203:000
204:000
205:543 REM
206:039 REM      Hier beginnt die Mouseabfrage
207:543 REM
208:000
209:016 choose:
210:345 COLOR 7,5
211:408 Anzahl=LOF(1)/104
212:467 LOCATE 26,61 :PRINT Anzahl" Datensätze";
213:951 LOCATE 26,5 :PRINT nr;" Datensatz";
214:950 WHILE MOUSE(0)=0
215:483 x= MOUSE (1)
216:997 y= MOUSE (2)
217:316 COLOR 1,0
218:886 LOCATE 26,30
219:192 IF y>156 AND y<180 THEN
220:276   ch=INT((x-12)/77)+2
221:123   IF x<40 THEN ch=1
222:846   IF ch=1 AND such=0 THEN PRINT"      Datensatz zurueck "
223:530   IF ch=1 AND such=1 THEN PRINT"      Suchmodus verlassen"
224:010   IF ch=2 THEN PRINT"      Datensatz weiter "
225:874   IF ch=3 THEN PRINT"      Datensatz aendern "
226:418   IF ch=4 THEN PRINT"      Datensatz suchen "
227:778   IF ch=5 THEN PRINT"      Datensatz eingeben "
228:602   IF ch=6 THEN PRINT"      Nummer sprechen "
229:338   IF ch=7 THEN PRINT"      Adresse Drucken "
230:306   IF ch=8 THEN PRINT"      Datenfile loeschen "
231:618   IF ch=9 THEN PRINT"      Raus hier !! "
232:000
233:091   ELSE
234:506     PRINT"      -- -- -- "
235:216   END IF
236:196 WEND
237:000
238:491 x=MOUSE(5)
239:005 y=MOUSE(6)
240:808 IF y>160 AND y<180 THEN
241:276   ch=INT((x-12)/77)+2
242:466   IF x<40 THEN ch=1 : such = 0
243:091 ELSE

```



```

244:994      ch=0
245:863      GOTO choose
246:216 END IF
247:948 WHILE MOUSE(0)<0
248:196 WEND
249:000
250:982 RETURN
251:000
252:000
253:012 rueck:
254:683 nr=nr-1
255:795 IF nr<=0 THEN nr=1
256:558 GET #1,nr
257:492 GOSUB uebertragen
258:258 GOSUB out
259:948 WHILE MOUSE(0)<0
260:196 WEND
261:982 RETURN
262:000
263:226 vor:
264:679 nr=nr+1
265:992 IF nr>Anzahl THEN nr=Anzahl
266:558 GET #1,nr
267:492 GOSUB uebertragen
268:258 GOSUB out
269:948 WHILE MOUSE(0)<0
270:196 WEND
271:000
272:982 RETURN
273:000
274:736 change:
275:000
276:340 COLOR 7,0
277:930 change=1:GOSUB out
278:316 COLOR 1,0
279:102 x=26
280:599 change=1:dummy$=na$:y=3:ln=20:GOSUB getstring
281:804 IF in$(x)="" THEN na$=in$
282:894 LOCATE 3,26:PRINT na$
283:871 change=1:dummy$=vn$:y=5:GOSUB getstring
284:732 IF in$(x)="" THEN vn$=in$
285:144 LOCATE 5,26:PRINT vn$
286:175 change=1:dummy$=st$:y=8:GOSUB getstring
287:732 IF in$(x)="" THEN st$=in$
288:432 LOCATE 8,26:PRINT st$
289:215 change=1:dummy$=pl$:y=10:ln=4:GOSUB getstring
290:284 IF in$(x)="" THEN pl$=in$
291:012 LOCATE 10,26:PRINT pl$
292:031 change=1:dummy$=ot$:y=12:ln=20:GOSUB getstring
293:476 IF in$(x)="" THEN ot$=in$
294:216 LOCATE 12,26:PRINT ot$
295:527 change=1:dummy$=tl$:y=15:GOSUB getstring
296:540 IF in$(x)="" THEN tl$=in$
297:508 LOCATE 15,26:PRINT tl$
298:760 GOSUB schreib
299:982 RETURN
300:000
301:966 watch:
302:316 COLOR 1,0
303:583 GOSUB Loeschmaske
304:554 LOCATE 26,30:PRINT " Kriterium wählen "
305:950 WHILE MOUSE(0)=0
306:483 x=MOUSE(1)
307:997 y=MOUSE(2)
308:200 IF x>195 AND x<364 THEN
309:915 wahl=INT(y/8)+1
310:216 END IF
311:196 WEND
312:491 x=MOUSE(5)
313:005 y=MOUSE(6)
314:200 IF x>195 AND x<364 THEN
315:915 wahl=INT(y/8)+1
316:091 ELSE
317:646 GOTO watch
318:216 END IF
319:000
320:000
321:839 IF wahl = 3 THEN x=26:y=3:ln=20:w=1
322:495 IF wahl = 5 THEN x=26:y=5:ln=20:w=1
323:479 IF wahl = 8 THEN x=26:y=8:ln=20:w=1
324:791 IF wahl = 10 THEN x=26:y=10:ln=4:w=1
325:567 IF wahl = 12 THEN x=26:y=12:ln=20:w=1
326:463 IF wahl = 15 THEN x=26:y=15:ln=20:w=1
327:000
328:990 IF w=0 THEN GOTO watch
329:518 w=0
330:000
331:204 GOSUB getstring : sr$=in$
332:122 nr=0
333:495 GOSUB lese
334:753 dummy=MOUSE(0)
335:982 RETURN
336:000
337:000
338:128 neu:
339:194 GOSUB in:
340:311 neu=1
341:650 GOSUB neuschreib:
342:982 RETURN
343:000
344:000
345:376 Talk:
346:513 FOR s=1 TO LEN(t1$)
347:539 y$=MID$(t1$,s,1)

```

```

348:121      GOSUB Translate
349:986      SAY Talk$
350:403 NEXT s
351:753 dummy=MOUSE(0)
352:982 RETURN
353:000
354:000
355:236 Translate:
356:386 Talk$=" "
357:406 IF y$="0" THEN Talk$="NUHL."
358:274 IF y$="1" THEN Talk$="AYNS."
359:090 IF y$="2" THEN Talk$="TSWAY."
360:338 IF y$="3" THEN Talk$="DRXAY."
361:614 IF y$="4" THEN Talk$="FIYRX."
362:502 IF y$="5" THEN Talk$="FIHAXNF."
363:546 IF y$="6" THEN Talk$="SEHKS."
364:934 IF y$="7" THEN Talk$="SIYBEHN."
365:606 IF y$="8" THEN Talk$="AH/CT."
366:750 IF y$="9" THEN Talk$="NOYN."
367:512 IF y$=" " OR y$="/" THEN FOR p=1 TO 100:NEXT p
368:982 RETURN
369:000
370:000
371:100 Dru:
372:511 CLOSE 1
373:210 GOSUB cut
374:000
375:454 LPRINT vn$ " na$
376:736 LPRINT st$ "
377:704 LPRINT pl$ " ot$
378:900 LPRINT
379:724 LPRINT tl$
380:900 LPRINT
381:900 LPRINT
382:201 GOSUB oeffne
383:982 RETURN
384:000
385:000
386:630 loesch:
387:320 COLOR 2,0
388:330 LOCATE 26,32:PRINT " Ganz sicher ??? "
389:022 BEEP
390:711 FOR pause=1 TO 3000:NEXT pause
391:586 LOCATE 26,30:PRINT "Noch einmal anwählen"
392:711 FOR pause=1 TO 3000:NEXT pause
393:514 LOCATE 26,30:PRINT "
394:983 GOSUB choose
395:766 IF ch<>8 THEN RETURN
396:511 CLOSE 1
397:630 KILL "df0:telefon.dat"
398:201 GOSUB oeffne
399:982 RETURN
400:000
401:000
402:396 raus:
403:503 CLS
404:511 CLOSE 1
405:500 END
406:000
407:000
408:968 Loeschmaske:
409:316 COLOR 1,0
410:474 a$=" "
411:374 LOCATE 3,26 :PRINT a$
412:470 LOCATE 5,26 :PRINT a$
413:614 LOCATE 8,26 :PRINT a$
414:114 LOCATE 10,26:PRINT "
415:014 LOCATE 12,26 :PRINT a$
416:158 LOCATE 15,26 :PRINT a$
417:650 LINE (454,32)-(596,118),0,bf
418:982 RETURN
419:000
420:000
421:000
422:506 in:
423:583 GOSUB Loeschmaske
424:336 COLOR 6,0
425:616 x=26:ln=20
426:604 y=3:GOSUB getstring:na$=in$
427:748 y=5:GOSUB getstring:vn$=in$
428:572 y=8:GOSUB getstring:st$=in$
429:848 y=10:ln=4:GOSUB getstring:pl$=in$:ln=20
430:876 y=12:GOSUB getstring:ot$=in$
431:764 y=15:GOSUB getstring:tl$=in$
432:753 dummy=MOUSE(0)
433:316 COLOR 1,0
434:258 GOSUB out
435:982 RETURN
436:000
437:000
438:102 cut:
439:784 FOR d=1 TO 20
440:708 IF ASC(MID$(vn$,d,1))>32 THEN NEXT d
441:225 vn$=LEFT$(vn$,d)
442:982 RETURN
443:000
444:522 neuschreib:
445:752 FOR f=1 TO Anzahl
446:730 GET #1,f
447:270 IF n$<na$ THEN NEXT f
448:144 nr=f
449:763 FOR f=Anzahl TO nr STEP -1
450:730 GET #1,f

```

```

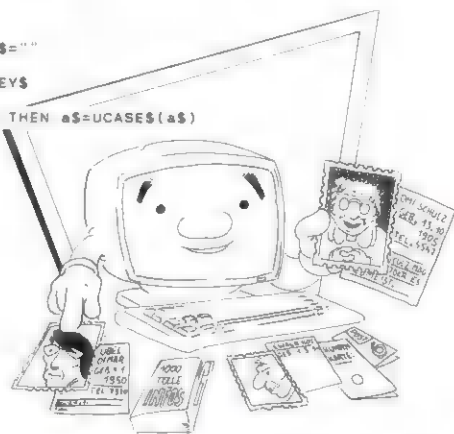
451:407 PUT #1,f+1
452:390 NEXT f
453:310 neu = 0
454:000
455:562 schreib:
456:000
457:406 LSET n$=na$
458:720 LSET v$=vn$
459:624 LSET s$=st$
460:500 LSET p$=pl$
461:480 LSET o$=ot$
462:644 LSET t$=tl$
463:734 PUT #1,nr
464:346 LOCATE 26,30:PRINT "Datensatz übernommen"
465:192 FOR pause = 1 TO 2000:NEXT
466:982 RETURN
467:000
468:000
469:198 out:
470:583 GOSUB Loeschmaske
471:222 IF change=1 THEN COLOR 7,0:change=0
472:894 LOCATE 3,26 :PRINT na$
473:144 LOCATE 5,26 :PRINT vn$
474:432 LOCATE 8,26 :PRINT st$
475:012 LOCATE 10,26:PRINT pl$
476:216 LOCATE 12,26 :PRINT ot$
477:508 LOCATE 15,26 :PRINT tl$
478:416 GOSUB Bild
479:982 RETURN
480:000
481:000
482:786 uebertragen:
483:260 na$=n$
484:740 vn$=v$
485:734 st$=s$
486:504 pl$=p$
487:598 ot$=o$
488:640 tl$=t$
489:982 RETURN
490:000
491:000
492:000
493:138 Bild:
494:543 tel = ASC ( LEFT$ (t$,1))
495:504 IF tel > 57 OR tel < 48 THEN
496:401 GOSUB Briefumschlag
497:091 ELSE
498:876 GOSUB telefon
499:216 END IF
500:982 RETURN
501:000
502:836 Briefumschlag:
503:530 LINE (460,40)-(590,90),1,bf
504:268 LINE (460,40)-(590,90),0
505:148 LINE (460,90)-(590,40),0
506:982 RETURN
507:000
508:794 telefon:
509:017 READ xpos,ypos
510:285 PSET (xpos+454,ypos+32),1
511:000
512:788 FOR d=1 TO 32
513:345 READ xpos
514:353 READ ypos
515:461 LINE -(xpos+454,ypos+32),1
516:160 NEXT
517:285 PAINT (510,40),1
518:005 RESTORE
519:000
520:982 RETURN
521:000
522:259 DATA 71,86,69,83,64,84,65,82,61,82,52,78,55,78,65,75
523:788 DATA 45,75,40,70,52,60,61,60,67,63,68,65,77,60
524:868 DATA 81,55,86,36,79,20,70,13,70,15,62,20,52,20
525:653 DATA 40,10,47, 5,60, 4,75, 9,88,18,95,27,95,45,91,53
526:635 DATA 80,67,70,72,65,75
527:000
528:296 lese:
529:828 WHILE Anzahl-nr>0
530:679 nr=nr+1
531:558 GET#1,nr
532:392 IF wahl=3 THEN
533:159 IF UCASE$(sr$)<>UCASE$(LEFT$(n$,LEN(sr$))) THEN GOT
534:216 END IF
535:424 IF wahl=5 THEN
536:887 IF UCASE$(sr$)<>UCASE$(LEFT$(v$,LEN(sr$))) THEN GOT
537:216 END IF
538:472 IF wahl=8 THEN
539:239 IF UCASE$(sr$)<>UCASE$(LEFT$(s$,LEN(sr$))) THEN GOT
540:216 END IF
541:312 IF wahl=10 THEN
542:591 IF UCASE$(sr$)<>UCASE$(LEFT$(p$,LEN(sr$))) THEN GOT
543:216 END IF
544:344 IF wahl=12 THEN
545:375 IF UCASE$(sr$)<>UCASE$(LEFT$(o$,LEN(sr$))) THEN GOT
546:216 END IF
547:392 IF wahl=15 THEN
548:679 IF INSTR(1,t$,sr$)=0 THEN GOTO lese
549:216 END IF
550:000

```

```

551:492 GOSUB uebertragen
552:482 Nummer=nr
553:258 GOSUB out
554:356 Marke3:
555:011 such=1
556:983 GOSUB choose
557:070 IF ch=1 THEN RETURN
558:000
559:392 GOSUB Auswahl
560:216 IF w<>1 THEN
561:316 COLOR 1,0
562:934 LOCATE 26,30:PRINT " Suchmodus verlassen ! "
563:022 BEEP
564:711 FOR pause=1 TO 3000:NEXT pause
565:322 LOCATE 26,30:PRINT "
566:000
567:033 GOTO Marke3
568:216 END IF
569:518 w=0
570:196 WEND
571:368 nr=Nummer
572:982 RETURN
573:000
574:000
575:826 Auswahl:
576:311 IF ch=6 THEN m=1:GOSUB Talk
577:633 IF ch=7 THEN m=1:GOSUB Dru
578:889 IF ch=10 THEN w=1:GOSUB raus
579:783 IF ch=2 THEN w=1
580:982 RETURN
581:000
582:000
583:000
584:176 getstring:
585:754 i=0:in$="":a$=""
586:352 Marke1:
587:615 LOCATE y,x+i
588:000
589:328 IF change=0 THEN
590:495 IF i<ln-1 THEN PRINT " _ ";
591:936 FOR p=1 TO ln-1-2
592:015 PRINT " ";
593:400 NEXT p
594:091 ELSE
595:316 COLOR 1,0
596:377 PRINT LEFT$(dummy$,1);
597:526 change =0
598:216 END IF
599:615 LOCATE y,x+i
600:000
601:628 up=1:GOSUB Taste:
602:776 IF ASC(a$)=13 THEN
603:615 LOCATE y,x+i
604:994 IF i<ln THEN PRINT " "
605:982 RETURN
606:000
607:216 END IF
608:680 IF ASC(a$)=8 THEN
609:728 IF LEN(in$)=>1 THEN
610:407 in$=LEFT$(in$,LEN(in$)-1)
611:674 a$=""
612:087 i=i-1
613:091 ELSE
614:443 d=1
615:216 END IF
616:216 END IF
617:046 IF i<ln THEN PRINT a$
618:066 in$=in$+a$
619:435 IF a$<>"" THEN i=i+1
620:464 IF i>ln THEN
621:087 i=i-1
622:159 IF a$<>"" THEN in$=LEFT$(in$,LEN(in$)-1)
623:216 END IF
624:638 IF d=1 THEN d=0:LINE (195,y*8-10)-(195,y*8),2
625:031 GOTO Marke1
626:000
627:924 Taste:
628:674 a$=""
629:066 WHILE a$=""
630:430 SLEEP
631:826 a$=INKEY$
632:196 WEND
633:077 IF up=0 THEN a$=UCASE$(a$)
634:170 up=0
635:982 RETURN

```



Listing 3: Dieses Adressenverwaltungsprogramm stellt die bisher erlernten und angewandten Möglichkeiten der Unterprogrammtechnik mit Hilfe von GOSUB-Anweisungen vor



Garry Glendown

## SprEd – das Sprite-Editor-Projekt

### Teil 4: Etwas fürs Auge

Diesmal wollen wir uns einer kleinen Nebensächlichkeits widmen: der Infoseite über das Programm SprEd. Und damit es auch wirklich gut aussieht, geben wir uns natürlich nicht nur mit einfachen Buchstaben zufrieden – unser Projekt muß schon eine ansprechende Grafik beinhalten.

**D**och wie es Grafiken zumeist an sich haben, ist auch die unsere keine der kleinsten. Bei 234 mal 132 Pixel ergibt sich für den erzeugten Source-Code eine Länge von etwa 30 kByte...

Ursprünglich wollten wir deshalb diese Datenwüste als Hexdump abdrucken, doch ergaben sich leider einige Schwierigkeiten bei der anschließenden Umwandlung, so daß wir auf den Source-Code leider nicht verzichten

konnten – man möge uns verzeihen...

### SprEd – es geht weiter

Nach der (zugegeben mühevollen) Eingabe des Listings 'Title.c' muß das File 'part2.c' aus unserer Ausgabe 3/90 (Seite 108) noch ein bißchen modifiziert werden. Löschen Sie dazu die Zeilen 61 und 62

```
61: DisplayAbout()
62: {}
```

oder dokumentieren Sie sie einfach aus (/.\*.\*/).

Danach kann man mit dem neuen Makefile SprEd compilieren.

In den nächsten Ausgaben wenden wir uns dem Farbrequenter und der Animation der einzelnen Sequenzen zu. Bis zum nächsten Mal also!

(br)

# Alle Includes wurden mit der Option '+hlib:all.pre' vorcompiliert

OBJ = spr.ed.o part2.o part3.o

```
title.o
.c.o:
cc $*.c +ilib:all.pre
spr.ed:
$(OBJ)
ln +q +cd $(OBJ) -la -lm -lc
```

### Wichtig!

Das Programm SprEd benötigt auf jeden Fall die Arp-Library. Diese ist Public Domain und darf deswegen nicht in unserem Magazin abgedruckt werden. Wenn Sie SprEd lauffähig machen wollen, sollten Sie sich zuerst die Arp-Library besorgen. Sie befindet sich unter anderem auf der Fishdisk 123 und auf der Panorama 27c. Bei eventuellen Fragen wenden Sie sich bitte an einen PD-Vertreiber.

### Listings

```
1: /***** Title.c *****/
2: ***
3: *** written in 1990 by Garry Glendown ***
4: *** (c) 1990 by DMV-Verlag ***
5: *** Sprache: C (Aztec 3.6) ***
6: *****/
7:
8: extern struct Screen *scr;
9:
10: static USHORT ImageData[] = {
11: 0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,
12: 0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,
13: 0x0000,0x001F,0xF818,0x019F,0xF818,0x019F,0xF818,0x019F,
14: 0x019F,0xF818,0x019F,0xF818,0x019F,0xF818,0x019F,0xF818,
15: 0xF818,0x0100,0x0180,0x1800,0x0180,0x0180,0x0180,0x0180,
16: 0x1800,0x0180,0x1800,0x0180,0x1800,0x0180,0x0180,0x1800,
17: 0x0180,0x1800,0x0100,0x0180,0x19FF,0x8180,0x19FF,0x8180,
18: 0x8180,0x19FF,0x8180,0x19FF,0x8180,0x19FF,0x8180,0x19FF,
19: 0x19FF,0x8180,0x19FF,0x8180,0x0000,0x0000,0x1801,0x8000,
20: 0x1801,0x8000,0x1801,0x8000,0x1801,0x8000,0x1801,0x8000,
21: 0x8000,0x1801,0x8000,0x1801,0x8000,0x1801,0x8000,0x1801,
22: 0x9FFF,0x8180,0x9FFF,0x8180,0x9FFF,0x8180,0x9FFF,0x8180,
23: 0x1801,0x9FFF,0x8180,0x9FFF,0x8180,0x9FFF,0x8180,0x0018,
24: 0x0001,0x8018,0x0000,0x8008,0x0000,0x8008,0x0000,0x0000,
25: 0x8008,0x0000,0x8018,0x0001,0x8018,0x0001,0x8000,0x0000,
26: 0x0019,0xFF81,0x8019,0xF800,0x0000,0x0000,0x0000,0x0000,
27: 0x0000,0x0000,0x0000,0x0000,0xFF81,0x8019,0xFF81,0x8019,
28: 0x8000,0x0018,0x0180,0x0018,0x0200,0x0000,0x0000,0x0000,
29: 0x0000,0x0000,0x0000,0x0000,0x0048,0x0180,0x0018,0x0018,
30: 0x0180,0x0000,0x3818,0x019F,0xF818,0x0180,0x0000,0x0000,
31: 0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,0x019F,
32: 0xF818,0x019F,0xF800,0x1800,0x0180,0x1800,0x0007F,0x0000,
33: 0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,
34: 0x0180,0x1800,0x0180,0x1800,0x19FF,0x8180,0x19FF,0x8180,
35: 0x8040,0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,
36: 0x03CF,0x8180,0x19FF,0x8180,0x1900,0x1801,0x8000,0x8000,
37: 0x1801,0x8040,0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,
38: 0x0000,0x03C1,0x8000,0x1801,0x8000,0x1800,0x1801,0x1801,
39: 0x9FFF,0x8180,0x9040,0x0000,0x0000,0x0000,0x0000,0x0000,
40: 0x0000,0x0000,0x03C1,0x9FFF,0x8180,0x1801,0x9FFF,0x8180,
41: 0x0001,0x8018,0x0001,0x8040,0x0000,0x0000,0x0000,0x0000,
42: 0x0000,0x0000,0x0000,0x03C1,0x8018,0x0001,0x8018,0x0018,
43: 0x0000,0x3F81,0x8019,0xFF81,0x8040,0x0000,0x0000,0x0000,
44: 0x0000,0x0000,0x0000,0x0000,0x03C1,0x8019,0xFF81,0x8019,
45: 0x8019,0xFF00,0x0180,0x0018,0x0180,0x0040,0x0000,0x0000,
46: 0x0000,0x0000,0x0000,0x0000,0x0000,0x03C0,0x0018,0x0018,
47: 0x0018,0x0018,0x0100,0x019F,0xF818,0x019F,0xF040,0x0000,
48: 0x07E6,0x0000,0x0000,0x01FF,0xFF80,0x07E0,0x03CF,0x0000,
49: 0xF818,0x019F,0xF818,0x0100,0x0180,0x1800,0x0180,0x0180,
50: 0x1840,0x181E,0x0000,0x0000,0x0078,0x0780,0x01E0,0x0000,
51: 0x03C0,0x1800,0x0180,0x1800,0x0100,0x0180,0x19FF,0x8180,
52: 0x8180,0x1040,0x7806,0x0000,0x0000,0x0078,0x0180,0x0000,
53: 0x01E0,0x03C0,0x19FF,0x8180,0x19FF,0x8180,0x0000,0x0000,
54: 0x1801,0x8000,0x1840,0x7800,0x0000,0x0000,0x0078,0x0180,
55: 0x0000,0x01E0,0x03C0,0x1801,0x8000,0x1801,0x8000,0x0000,
56: 0x1FF8,0x1801,0x9FFF,0x8180,0x1040,0x7E00,0x01F9,0xF818,
57: 0x7878,0x0000,0x7FE0,0x03C8,0x1801,0x9FFF,0x8180,0x1801,
58: 0x9F00,0x0018,0x0001,0x8018,0x0040,0x1FE0,0x007E,0x0000,
59: 0x7E1E,0x7878,0x0601,0x01E0,0x03C8,0x0001,0x8018,0x0001,
60: 0x8000,0x0019,0xFF81,0x8019,0xF040,0x07F8,0x0180,0x0180,
```

Listing: Title.c

```
61: 0x0078,0x1E1F,0x807F,0xFE07,0x81E0,0x03C9,0xFF81,0x8180,
62: 0x8019,0xFF81,0x8000,0x0018,0x0180,0x0018,0x0040,0x0000,
63: 0x007E,0x0078,0x1E1E,0x0078,0x0607,0x81E0,0x03C8,0x0180,
64: 0x0180,0x0018,0x0180,0x0000,0x3818,0x019F,0xF818,0x0040,
65: 0x001F,0x8078,0x1E1E,0x0078,0x0007,0x81E0,0x03C8,0x019F,
66: 0xF818,0x019F,0xF818,0x019F,0xF800,0x1800,0x0180,0x0180,
67: 0x1800,0x0040,0x0007,0x8078,0x1E1E,0x0078,0x0007,0x81E0,
68: 0x03C0,0x0180,0x0180,0x1800,0x0180,0x1800,0x19FF,0x8180,
69: 0x19FF,0x8040,0x6007,0x8078,0x1E1E,0x0078,0x0187,0x81E0,
70: 0x03CF,0x8180,0x19FF,0x8180,0x1900,0x1801,0x8000,0x1801,
71: 0x8040,0x781E,0x007E,0x781E,0x0078,0x0781,0xE0,0x03C1,0x8000,
72: 0x1801,0x8000,0x1801,0x8040,0x781E,0x007E,0x781E,0x0078,
73: 0x0781,0xE0,0x03C1,0x8000,0x1801,0x8000,0x1801,0x8000,0x1801,
74: 0x9FFF,0x8180,0x1801,0x9040,0x67E0,0x007F,0x8180,0x9FFF,
75: 0xFF80,0x7F80,0x0001,0x8018,0x0001,0x8040,0x0000,0x0078,
76: 0x0000,0x0000,0x0000,0x0000,0x03C1,0x8018,0x0001,0x8018,
77: 0x0000,0x0000,0x0000,0x3F81,0x8019,0xFF81,0x8040,0x0000,
78: 0x0078,0x0000,0x0000,0x0000,0x0000,0x0000,0x03C1,0x8019,
79: 0xFF81,0x8019,0xFF00,0x0180,0x0180,0x0018,0x0018,0x0040,
80: 0x0000,0x0078,0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,
81: 0x03C0,0x0018,0x0180,0x0018,0x0100,0x019F,0xF818,0x019F,
82: 0xF040,0x0000,0x01FE,0x0000,0x0000,0x0000,0x0000,0x0000,
83: 0x0000,0x03CF,0xF818,0x019F,0xF818,0x0100,0x0180,0x0180,
84: 0x1800,0x0180,0x1840,0x0000,0x0000,0x0000,0x0000,0x0000,
85: 0x0000,0x0000,0x03C0,0x1800,0x0180,0x1800,0x0100,0x0180,
86: 0x19FF,0x8180,0x1040,0x0000,0x0000,0x0000,0x0000,0x0000,
87: 0x0000,0x0000,0x0000,0x03C0,0x19FF,0x8180,0x19FF,0x8180,
88: 0x1000,0x0000,0x1801,0x8000,0x1840,0x0000,0x0000,0x0000,
89: 0x0000,0x0000,0x0000,0x0000,0x03C0,0x1801,0x8000,0x1801,
90: 0x8000,0x1FF8,0x1801,0x9FFF,0x8180,0x1040,0x0000,0x0000,
91: 0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,0x03C8,0x1801,
92: 0x9FFF,0x8180,0x9F00,0x0018,0x0001,0x8018,0x0040,0x0000,
93: 0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,0x03C8,0x0001,
94: 0x8018,0x0001,0x8000,0x0019,0xFF81,0x8019,0xF1FF,0xFFFF,
95: 0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,
96: 0xFCC9,0xFF81,0x8019,0xFF81,0x8000,0x0018,0x0180,0x0018,
97: 0x03FF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,
98: 0xFFFF,0xFF08,0x0180,0x0018,0x0180,0x0000,0x3818,0x019F,
99: 0xF818,0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,
100: 0x0000,0x0000,0x0018,0x019F,0xF818,0x019F,0xF800,0x1800,
101: 0x0180,0x1800,0x0000,0x0000,0x0000,0x0180,0x1800,0x0180,
102: 0x0000,0x0000,0x0000,0x0000,0x0180,0x1800,0x19FF,0x8180,
103: 0x1800,0x19FF,0x8180,0x19FF,0x8180,0x19FF,0x8180,0x19FF,
104: 0x8180,0x19FF,0x8180,0x19FF,0x8180,0x19FF,0x8180,0x19FF,
105: 0x8180,0x1900,0x1801,0x8000,0x1801,0x8000,0x1801,0x8000,
106: 0x8000,0x1801,0x8000,0x1801,0x8000,0x1801,0x8000,0x8000,
107: 0x1801,0x8000,0x1800,0x1801,0x9FFF,0x8180,0x1801,0x9FFF,
108: 0x8180,0x1801,0x9FFF,0x8180,0x9FFF,0x8180,0x9FFF,0x8180,
109: 0x9FFF,0x8180,0x9FFF,0x8180,0x0001,0x8018,0x0001,0x8018,
110: 0x0001,0x8018,0x0001,0x8018,0x0000,0x0000,0x3F81,0x8019,
111: 0xFF81,0x8019,0x0001,0x8019,0xF001,0x8000,0x7F00,0x0000,
112: 0x0000,0x0781,0x8019,0xFF81,0x8019,0xF000,0x0180,0x0018,
113: 0x0018,0x0018,0x0018,0x0000,0x0018,0x0000,0x0000,0x0000,
114: 0x0000,0x0000,0x0180,0x0018,0x0018,0x0018,0x0018,0x0018,
115: 0x019F,0xF818,0x019F,0xF800,0xFFC0,0x1818,0x0F07,0xC3FC,
116: 0x00FF,0x003F,0xC09F,0xF818,0x019F,0xF818,0x0100,0x0180,
117: 0x0180,0x0180,0x0180,0x0180,0x0180,0x003C,0x0800,0x3F00,
118: 0x0F3F,0x03CF,0xC0F0,0xF000,0x1800,0x0180,0x1800,0x0100,
119: 0x0180,0x0180,0x0180,0x19FF,0x8180,0x1930,0x0003,0x01FC,
120: 0xFF00,0x3C0F,0x0F03,0xC0F0,0xF000,0x19FF,0x8180,0x19FF,
121: 0x8180,0x0000,0x0000,0x0000,0x1801,0x8000,0x1830,0x3FC3,
122: 0x0000,0x0F00,0x3C03,0xCF00,0xF3C0,0x3C00,
```

Listing: Title.c

# Listing

```

124: 0x1801,0x8000,0x1801,0x8000,0x1FF8,0x1801,0x9FF8,
125: 0x1800,0xC0C0,0xC000,0x0F00,0x3C03,0xC000,0xF3C0,
126: 0x3C18,0x1801,0x9FF8,0x1801,0x9F00,0x0018,0x0001,
127: 0x8018,0x00C3,0x0000,0x0001,0x0F00,0x3C03,0xC000,
128: 0xF3C1,0x3C18,0x0001,0x8018,0x0001,0x8000,0x0019,
129: 0xFF81,0x8019,0xFFC3,0x0000,0xC181,0x8F01,0x3F03,
130: 0xCFC0,0xF3C1,0x3C19,0xFF81,0x8019,0xFF81,0x8000,
131: 0x0018,0x0180,0x0018,0x00C3,0x0000,0xC180,0x0F00,
132: 0x0FFF,0xC3FF,0xF3C0,0x3C18,0x0180,0x0018,0x0180,
133: 0x0000,0x3818,0x019F,0xF818,0x00C0,0xC0C0,0xC19F,
134: 0xC000,0x03CF,0x00F3,0xC3C1,0x3C18,0x019F,0xF818,
135: 0x019F,0xF800,0x1800,0x0180,0x1800,0x00D0,0x3F03,
136: 0x0180,0x0F00,0x000F,0x0003,0xC3C0,0x3C00,0x0180,
137: 0x1800,0x0180,0x1800,0x19FF,0x8180,0x19FF,0x8030,
138: 0x0003,0x0180,0x0F01,0x803C,0x000F,0x00F0,0xF01F,
139: 0x8180,0x19FF,0x8180,0x1900,0x1801,0x8000,0x1801,
140: 0x800F,0x003C,0x0000,0x3F00,0x83F0,0x00FC,0x00F0,
141: 0xF001,0x8000,0x1801,0x8000,0x1800,0x1801,0x9FF8,
142: 0x1801,0x9F80,0xFFC0,0x00FF,0xFF00,0x3F00,0x0FC0,
143: 0x043F,0xC001,0x9FF8,0x1801,0x9FF8,0x1800,0x0001,
144: 0x8018,0x0001,0x8000,0x0000,0x0018,0x0000,0x0000,
145: 0x0000,0x0000,0x0001,0x8018,0x0001,0x8018,0x0000,
146: 0x3F81,0x8019,0xFF81,0x8010,0x0000,0x0018,0x0000,
147: 0x8001,0xE000,0x0000,0x0181,0x8019,0xFF81,0x8019,
148: 0xFF00,0x0180,0x0018,0x0180,0x0018,0x0000,0x0018,
149: 0x0000,0x0018,0x0000,0x0000,0x0180,0x0180,0x0180,
150: 0x0100,0x019F,0xF818,0x019F,0xF818,0x019F,0xF818,
151: 0xF818,0x019F,0xF818,0x019F,0xF818,0x019F,0xF818,
152: 0x019F,0xF818,0x0100,0x0180,0x1800,0x0000,0x1800,
153: 0x0180,0x1800,0x0180,0x1800,0x0000,0x0000,0x0000,
154: 0x1800,0x0000,0x1800,0x0100,0x0180,0x19FF,0x8000,
155: 0x19FF,0x8180,0x1007,0x8180,0x19FF,0x8000,0x00FF,
156: 0x8000,0x09E0,0x0000,0x19FF,0x8180,0x0000,0x1801,
157: 0x87C0,0x1801,0x8000,0x07F1,0x8000,0x1801,0x9FF8,
158: 0xF000,0x07FF,0x0007,0xFFC0,0x1801,0x8000,0x1FF8,
159: 0x1801,0x9FF0,0x1801,0x9FF8,0x07F1,0x9FF8,0x1801,
160: 0x9FF8,0xFF00,0x7FFF,0xF01F,0xFFC8,0x1801,0x9F00,
161: 0x0018,0x0001,0x9FF0,0x0000,0x0000,0x07F1,0x8000,
162: 0x0001,0x87F1,0xFFC1,0xFFD7,0xFC7F,0x1FC8,0x0001,
163: 0x8000,0x0018,0xFF81,0x1FF1,0xC000,0x0000,0xC001,
164: 0x8000,0x0000,0x87F0,0x1FC1,0xF0C0,0x07C7,0x07C9,
165: 0xFF81,0x8000,0x0018,0x0180,0x7FFC,0x1FF7,0xF1FC,
166: 0x1FF0,0x1FF8,0x1FFC,0x07F0,0x1FF7,0xFC19,0xFF7F,
167: 0xC018,0x0180,0x0000,0x3818,0x019C,0x7FFC,0x1FF8,
168: 0xFFFF,0x1FF0,0x7FFF,0x7FFF,0x27F0,0x07F7,0xF018,
169: 0x77F7,0xFC18,0x019F,0xF800,0x1800,0x0181,0xF0DF,
170: 0x07FF,0xFFFF,0x07F1,0xFFFC,0x77F7,0x07F0,0x07F7,
171: 0xF000,0x7F1F,0xFF00,0x0180,0x0180,0x19FF,0x8181,
172: 0xF0DF,0x07FD,0xFF7F,0x07F1,0xFC7C,0x7FFF,0x07F3,
173: 0x87F7,0xF1FE,0x7F07,0xFFC7,0x8180,0x1900,0x1801,
174: 0x8001,0xFFFF,0x07F1,0xFC7F,0x07F1,0xF0FC,0x1FFF,
175: 0x07F1,0x87F7,0xF000,0x7F00,0x7FF1,0x8000,0x1800,
176: 0x1801,0x9FE7,0xFFFF,0xC7F1,0xFC7F,0x07F1,0xFFFC,
177: 0x7FFF,0x07F0,0x1FF7,0xF0C1,0xFF00,0x1FF1,0x9FF8,
178: 0x1800,0x0001,0x8007,0xF0F7,0xC7F1,0xFC7F,0x07F0,
179: 0x7FFF,0x77F7,0x07F0,0x1FC1,0xF0C1,0xFC7C,0x07F1,
180: 0x8018,0x0000,0x3F81,0x801F,0xC01F,0xF7F1,0xFC7F,
181: 0x07F0,0x7FFC,0x7FFF,0x07F1,0xFFC9,0xFF07,0xFC7F,
182: 0x1FF1,0x8019,0xFF00,0x0180,0x0007,0xF0F7,0xFFC7,
183: 0xFFFF,0xDFFC,0x7FFF,0xFFFF,0xDFFC,0xFF18,0x7FFF,
184: 0xF0F7,0xFFC0,0x0018,0x0100,0x019F,0xF87F,0xF0F7,
185: 0xFFFF,0xFFFF,0xDFF0,0xFFFF,0xDFF0,0xDFF0,0xFC18,
186: 0x07FF,0x007F,0xF0C0,0xF818,0x0100,0x0180,0x1800,
187: 0x0100,0x0000,0x0000,0x0001,0xF0C1,0x0000,0x0000,
188: 0x0000,0x0000,0x8000,0x0000,0x0000,0x0100,0x0180,
189: 0x1980,0x0180,0x0000,0x0000,0x0001,0xFF1F,0xC020,
190: 0x0000,0x01FF,0x8000,0x1980,0x0180,0x19FF,0x8100,
191: 0x0000,0x1801,0x8000,0x1801,0x8000,0x1801,0xFFFF,
192: 0xC801,0x8000,0x1801,0x8000,0x1801,0x8000,0x1801,
193: 0x8000,0x1FF8,0x1801,0x9FF8,0x1801,0x9FF8,0x1800,
194: 0x7FFF,0x1801,0x9FF8,0x1801,0x9FF8,0x1801,0x9FF8,
195: 0x1801,0x9F00,0x0018,0x0001,0x8018,0x0001,0x8018,
196: 0x0000,0x0000,0x0001,0x8018,0x0001,0x8018,0x0001,
197: 0x8018,0x0001,0x8000,0x0019,0xFF81,0x8019,0xFF81,
198: 0x8019,0xFF81,0x8000,0xFF81,0x8019,0xFF81,0x8019,
199: 0xFF81,0x8019,0xFF81,0x8000,0x0018,0x0180,0x0018,
200: 0x0180,0x0018,0x0180,0x0018,0x0180,0x0018,0x0180,
201: 0x0018,0x0180,0x0018,0x0180,0x0180,0x0180,0x019F,
202: 0xF818,0x019F,0xF818,0x019F,0xF818,0x019F,0xF818,
203: 0x019F,0xF818,0x019F,0xF818,0x019F,0xF800,0x1800,
204: 0x0180,0x1800,0x0180,0x1800,0x0180,0x1800,0x0180,
205: 0x1800,0x0180,0x1800,0x0180,0x1800,0x0180,0x1800,
206: 0x19FF,0x8180,0x19FF,0x8180,0x19FF,0x8180,0x19FF,
207: 0x0180,0x19FF,0x8180,0x19FF,0x8180,0x19FF,0x8180,
208: 0x1900,0x1801,0x8000,0x1801,0x8000,0x1801,0x8000,
209: 0x1800,0xE000,0x1801,0x8000,0x1801,0x8000,0x1801,
210: 0x8000,0x1800,0x1801,0x9FF8,0x1801,0x9FF8,0x1801,
211: 0x9FF8,0x180F,0x39F8,0x1801,0x9FF8,0x1801,0x9FF8,
212: 0x1801,0x9FF8,0x1800,0x0001,0x8018,0x0001,0x8018,
213: 0x0001,0x8018,0x0000,0x3818,0x0001,0x8018,0x0001,
214: 0x8018,0x0001,0x8018,0x0000,0x3F81,0x8019,0xFF81,
215: 0x8019,0xFF81,0x8019,0xFF00,0x3801,0xFF81,0x8019,
216: 0xFF81,0x8019,0xFF81,0x8019,0xFF00,0x0180,0x0018,
217: 0x0180,0x0018,0x0180,0x0018,0x0000,0xF800,0x0180,
218: 0x0018,0x0180,0x0018,0x0180,0x0018,0x0100,0x019F,
219: 0xF818,0x019F,0xF818,0x019F,0xF818,0x00C3,0xC038,
220: 0x019F,0xF818,0x019F,0xF818,0x019F,0xF818,0x0100,
221: 0x0180,0x1800,0x0180,0x1800,0x0180,0x1800,0x0003,
222: 0xB0F8,0x0180,0x1800,0x0180,0x1800,0x0180,0x1800,
223: 0x0100,0x0180,0x19FF,0x8180,0x19FF,0x8180,0x19FF,
224: 0x8030,0xC3E3,0x8180,0x19FF,0x8180,0x19FF,0x8180,
225: 0x19FF,0x8100,0x0000,0x1801,0x8000,0x1801,0x8000,
226: 0x1801,0x80F0,0xF818,0x8000,0x1801,0x8000,0x1801,
227: 0x8000,0x1801,0x8000,0x1FF8,0x1801,0x9FF8,0x1801,
228: 0x9FF8,0x1801,0x9FEC,0x0E01,0x9FF8,0x1801,0x9FF8,
229: 0x1801,0x9FF8,0x1801,0x9F00,0x0018,0x0001,0x8018,

```

Listing: Title.c

```

230: 0x0001,0x8018,0x0001,0x80E0,0x0380,0x8018,0x0001,
231: 0x8018,0x0001,0x8018,0x0001,0x8000,0x0019,0xFF81,
232: 0x8019,0xFF81,0x8019,0xFF81,0x8030,0xC00E,0x0019,
233: 0xFF81,0x8019,0xFF81,0x8019,0xFF81,0x8000,0x0018,
234: 0x0180,0x0018,0x0180,0x0018,0x0018,0x00C3,0xF03E,
235: 0x0018,0x0180,0x0018,0x0180,0x0018,0x0180,0x0000,
236: 0x3818,0x019F,0xF818,0x019F,0xF818,0x019F,0x23FF,
237: 0x8FF8,0xF818,0x019F,0xF818,0x019F,0xF818,0x019F,
238: 0xF800,0x1800,0x0180,0x1800,0x0180,0x1800,0x0180,
239: 0x1000,0x0000,0x1800,0x0180,0x1800,0x0180,0x1800,
240: 0x0180,0x1800,0x19FF,0x8180,0x19FF,0x8180,0x19FF,
241: 0x8180,0x1800,0x0000,0x19FF,0x8180,0x19FF,0x8180,
242: 0x19FF,0x8180,0x1900,0x1801,0x8000,0x0001,0x8000,
243: 0x1800,0x8000,0x1801,0x8000,0x1800,0x1800,0x0000,0x1801,
244: 0x8000,0x1801,0x8000,0x1800,0x1801,0x8000,0x0001,
245: 0x9FF8,0x0000,0x0100,0x1801,0x9FF8,0x1800,0x0078,
246: 0x1801,0x9FF8,0x1801,0x9FF8,0x1800,0x0000,0x0078,
247: 0x6001,0x8018,0x07F8,0x607E,0x0001,0x8018,0x0001,
248: 0xF818,0x0001,0x8018,0x0001,0x8018,0x0000,0x3F80,
249: 0x7807,0xE081,0x8018,0x7807,0xE01E,0xF818,0x8019,
250: 0xFF80,0x7819,0xFF81,0x8019,0xFF81,0x8019,0xFF00,
251: 0x0181,0xE001,0xE080,0x0019,0xE001,0xE01E,0x0180,
252: 0x0000,0x0180,0x7818,0x0080,0x0000,0x0000,0x0018,
253: 0x0100,0x0191,0x0000,0x609F,0xF811,0xE000,0x001E,
254: 0x0000,0x0000,0x0080,0x7800,0x0000,0x0000,0x0000,
255: 0x0018,0x0100,0x0187,0x8000,0x0080,0x1807,0x8080,
256: 0x001E,0x007F,0x87E1,0xE01F,0xF807,0xF81F,0x87E1,
257: 0xFFE1,0xE000,0x0100,0x0187,0x81F0,0x0000,0x19E7,
258: 0x8180,0x001E,0x01E1,0xE1E7,0xF878,0x781E,0x1E07,
259: 0x8781,0xE1E7,0xF83F,0x8100,0x0007,0x8007,0xF800,
260: 0x1807,0x8007,0xF81E,0x0780,0x61F8,0x79E0,0x7878,
261: 0x0787,0x8187,0x81F8,0x7801,0x8000,0x1FE7,0x8001,
262: 0xE038,0x1807,0x83E1,0xE01E,0x07FF,0xE1E0,0x79E0,
263: 0x7878,0x0781,0x7878,0x81E0,0x7801,0x9F00,0x0007,
264: 0x8001,0xE018,0x0007,0x8011,0xE01E,0x0780,0x01E0,
265: 0x79E0,0x7878,0x0781,0xE7E7,0x81E0,0x7801,0x8000,
266: 0x0001,0xE001,0xE000,0xFF81,0xE001,0xE080,0x0780,
267: 0x01E0,0x79E0,0x7878,0x0780,0x7E65,0x01E0,0x7801,
268: 0x8000,0x0011,0xE001,0xE000,0x0181,0xE001,0xE09E,
269: 0x0780,0x01E0,0x79E0,0x7878,0x0780,0x7E7E,0x01E0,
270: 0x7800,0x0000,0x3818,0x7E1F,0x8078,0x0198,0x7E1F,
271: 0x801E,0x01E1,0xE1E0,0x7878,0x781E,0x1E00,0x1E18,
272: 0x01E0,0x781F,0xF800,0x1800,0x07F8,0x0078,0x0180,
273: 0x07F8,0x007F,0x807F,0x87F8,0x7E1F,0xFE07,0xF800,
274: 0x1818,0x07F8,0x7E00,0x1800,0x19FE,0x0000,0x0000,
275: 0x0180,0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,
276: 0x000F,0x8000,0x0000,0x0000,0x1800,0x1801,0x8000,
277: 0x0800,0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,
278: 0x0000,0x0001,0x8000,0x1000,0x0000,0x1800,0x1801,
279: 0x9800,0x1800,0x9FF8,0x1800,0x9F80,0x0800,0x0800,
280: 0x0000,0x0038,0x0001,0x80E0,0x1800,0x8038,0x1800,
281: 0x0001,0x8018,0x0001,0x8018,0x0001,0x8018,0x0001,
282: 0x8018,0x0001,0x8018,0x0001,0x8018,0x0001,0x8018,
283: 0x0000,0x3F81,0x8019,0xFF81,0x8019,0xFF81,0x8019,
284: 0xFF81,0x8019,0xFF81,0x8019,0xFF81,0x8019,0xFF81,
285: 0x8019,0xFF00,0x0180,0x0018,0x0018,0x0018,0x0180,
286: 0x0018,0x0180,0x0018,0x0018,0x0018,0x0018,0x0018,
287: 0x019F,0xF818,0x019F,0xF818,0x019F,0xF818,0x019F,
288: 0xF818,0x019F,0xF818,0x0100,0x0180,0x1800,0x0180,
289: 0x1800,0x0180,0x0180,0x0180,0x1800,0x0180,0x1800,
290: 0x0180,0x1800,0x0180,0x1800,0x0100,0x0000,0x0000,
291: 0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,
292: 0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,
293: 0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,0xFFFF,
294: 0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,
295: 0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,
296: 0xC19F,0xF818,0x019F,0xF818,0x019F,0xF818,0x019F,
297: 0xF818,0x019F,0xF818,0x019F,0xF818,0x019F,0xF818,
298: 0x01C0,0xC180,0x1800,0x0180,0x1800,0x0180,0x1800,
299: 0x0180,0x1800,0x0180,0x1800,0x0180,0x1800,0x0180,
300: 0x1800,0x01C0,0xC180,0x19FF,0x8180,0x19FF,0x8180,
301: 0x19FF,0x8180,0x19FF,0x8180,0x19FF,0x8180,0x19FF,
302: 0x8180,0x19FF,0x81C0,0xC000,0x1801,0x8000,0x1801,
303: 0x8000,0x1801,0x8000,0x1801,0x8000,0x1801,0x8000,
304: 0x1801,0x8000,0x1801,0x80C0,0x0FF8,0x9801,0x9FF8,
305: 0x1801,0x9FF8,0x1801,0x9FF8,0x1801,0x9FF8,0x1801,
306: 0x9FF8,0x1801,0x9FF8,0x1801,0x9F00,0xC018,0x0001,
307: 0x8018,0x0000,0x8008,0x0000,0x8008,0x0000,0x8008,
308: 0x0000,0x8018,0x0001,0x8018,0x0001,0x80C0,0xC019,
309: 0xFF81,0x8019,0xF800,0x0000,0x0000,0x0000,0x0000,
310: 0x0000,0x0000,0x0019,0xFF81,0x8019,0xFF81,0x80C0,
311: 0xC018,0x0180,0x0018,0x0355,0x5555,0x5555,0x5555,
312: 0x5555,0x5555,0x5555,0x5555,0x5555,0x0018,0x0180,
313: 0x00C0,0xF818,0x019F,0xF818,0x03AA,0xAAAA,0xAAAA,
314: 0xAAAA,0xAAAA,0xAAAA,0xAAAA,0xAAC8,0x019F,0xF818,
315: 0x019F,0xF8C0,0x0800,0x0180,0x1800,0x017F,0xFFFF,
316: 0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,
317: 0x1800,0x0180,0x1800,0x09FF,0x8180,0x19FF,0x82FF,
318: 0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,
319: 0x8180,0x19FF,0x8180,0x19C0,0x0801,0x8000,0x1801,
320: 0x817F,0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,
321: 0xFFC1,0x8000,0x1801,0x8000,0x18C0,0x0801,0x9FF8,
322: 0x1801,0x92FF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,
323: 0xFFFF,0xFFC1,0x9FF8,0x1801,0x9FF8,0x18C0,0xC001,
324: 0x8018,0x0001,0x817F,0xFFFF,0xFFFF,0xFFFF,0xFFFF,
325: 0xFFFF,0xFFFF,0xFFC1,0x8018,0x0001,0x8018,0x00C0,
326: 0xFF81,0x8019,0xFF81,0x82FF,0xFFFF,0xFFFF,0xFFFF,
327: 0xFFFF,0xFFFF,0xFFFF,0xFFC1,0x8019,0xFF81,0x8019,
328: 0xFFC0,0xC180,0x0018,0x0180,0x017F,0xFFFF,0xFFFF,
329: 0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFC0,0x0018,
330: 0x0018,0x01C0,0xC19F,0x81F8,0x019F,0x22FF,0xF800,
331: 0x3FFF,0xFFFF,0xFE00,0x000F,0xF803,0xFFC0,0xF818,
332: 0x019F,0xF818,0x01C0,0xC180,0x1800,0x0180,0x19FF,
333: 0xE000,0x3FFF,0xFFFF,0xFE00,0x000F,0xF803,0xFFC0,
334: 0x1800,0x0180,0x1800,0x01C0,0xC180,0x19FF,0x8180,
335: 0x12FF,0x80E0,0x3FFF,0xFFFF,0xFFFF,0xFF80,0xF80F,0xFE03,

```

Listing: Title.c



```

336: 0xFFC0,0x19FF,0x8180,0x19FF,0x81C0,0xC000,0x1801,
337: 0x8000,0x197F,0x80F8,0x3FFF,0xFFFF,0xFF80,0xF80F,
338: 0xF803,0xFFC0,0x1801,0x8000,0x1801,0x80C0,0xDFF8,
339: 0x1801,0x9FF8,0x12FF,0x803F,0xF800,0x0080,0x0080,
340: 0xFFFF,0x8003,0xFFC8,0x1801,0x9FF8,0x1801,0x9FC0,
341: 0xC018,0x0001,0x8018,0x017F,0x8003,0xF800,0x0000,
342: 0x0080,0xF83E,0x0003,0xFFC8,0x0001,0x8018,0x0001,
343: 0x80C0,0xC019,0xFF81,0x8019,0xF2FF,0xE000,0xFF80,
344: 0x0020,0x0080,0x0038,0x0203,0xFFC9,0xFF81,0x8019,
345: 0xFF81,0x80C0,0xC018,0x0180,0x0018,0x017F,0xF800,
346: 0x3F80,0xE020,0x0F80,0x0038,0x0E03,0xFFC8,0x0180,
347: 0x0018,0x0180,0x00C0,0xF818,0x019F,0xF818,0x02FF,
348: 0xFF80,0x0F80,0xE020,0x3F80,0xF838,0x0E03,0xFFC8,
349: 0x019F,0xF818,0x019F,0xF8C0,0xD800,0x0180,0x1800,
350: 0x017F,0xFFE0,0x0F80,0xE020,0x3F80,0xFF80,0x0E03,
351: 0xFFC0,0x0180,0x1800,0x0180,0x18C0,0x09FF,0x8180,
352: 0x19FF,0x82FF,0x83F8,0x0F80,0xE020,0x3F80,0xF808,
353: 0x0E03,0xFFC0,0x8180,0x19FF,0x8180,0x19C0,0xD801,
354: 0x8000,0x1801,0x817F,0x80E0,0x0F80,0x0020,0x3F80,
355: 0xF808,0x0203,0xFFC1,0x8000,0x1801,0x8000,0x18C0,
356: 0xD801,0x9FF8,0x1801,0x92FF,0x8000,0x3F80,0x0180,
357: 0xE000,0x000E,0x0000,0xFFC1,0x9FF8,0x1801,0x9FF8,
358: 0x18C0,0xC001,0x8018,0x0001,0x817F,0x8003,0xFF80,
359: 0x0380,0xE000,0x000F,0x3F80,0xFFC1,0x8018,0x0001,
360: 0x8018,0x00C0,0xFF81,0x8019,0xFF81,0x82FF,0xFFFF,
361: 0xFF80,0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFC1,0x8019,
362: 0xFF81,0x8019,0xFFC0,0xC180,0x0018,0x0180,0x017F,
363: 0xFFFF,0xFF80,0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFC0,
364: 0x0018,0x0180,0x0018,0x01C0,0xC19F,0xF818,0x019F,
365: 0xF2FF,0xFFFF,0xF800,0x3FFF,0xFFFF,0xFFFF,0xFFFF,
366: 0xFFC0,0xF818,0x019F,0xF818,0x01C0,0xC180,0x1800,
367: 0x0180,0x197F,0xFFFF,0xF800,0x3FFF,0xFFFF,0xFFFF,
368: 0xFFFF,0xFFC0,0x1800,0x0180,0x1800,0x01C0,0xC180,
369: 0x19FF,0x8180,0x12FF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,
370: 0xFFFF,0xFFFF,0xFFC0,0x19FF,0x8180,0x19FF,0x81C0,
371: 0xC000,0x1801,0x8000,0x197F,0xFFFF,0xFFFF,0xFFFF,
372: 0xFFFF,0xFFFF,0xFFFF,0xFFC0,0x1801,0x8000,0x1801,
373: 0x80C0,0xDFF8,0x1801,0x9FF8,0x12FF,0xFFFF,0xFFFF,
374: 0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFC8,0x1801,0x9FF8,
375: 0x1801,0x9FC0,0xC018,0x0001,0x8018,0x017F,0x9FF8,
376: 0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFC8,0x0001,
377: 0x8018,0x0001,0x80C0,0xC019,0xFF81,0x8019,0x3FFF,
378: 0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFC9,
379: 0xFF81,0x8019,0xFF81,0x80C0,0xC018,0x0180,0x0018,
380: 0x03FF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,
381: 0xFFC8,0x0180,0x0018,0x0180,0x00C0,0xF818,0x019F,

```

Listing: Title.c

```

382: 0xF818,0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,
383: 0x0000,0x0018,0x019F,0xF818,0x019F,0xF8C0,0xD800,
384: 0x0180,0x1800,0x0000,0x0000,0x0000,0x0000,0x0000,
385: 0x0000,0x0000,0x0000,0x0180,0x1800,0x0180,0x18C0,
386: 0xD9FF,0x8180,0x19FF,0x8180,0x19FF,0x8180,0x19FF,
387: 0x8180,0x19FF,0x8180,0x19FF,0x8180,0x19FF,0x8180,
388: 0x19C0,0xD801,0x8000,0x1801,0x8000,0x1801,0x8000,
389: 0x1801,0x8000,0x1801,0x8000,0x1801,0x8000,0x1801,
390: 0x8000,0x18C0,0xD801,0x9FF8,0x1801,0x9FF8,0x1801,
391: 0x9FF8,0x1801,0x9FF8,0x1801,0x9FF8,0x1801,0x9FF8,
392: 0x1801,0x9FF8,0x18C0,0xC001,0x8018,0x0001,0x8018,
393: 0x0001,0x8018,0x0001,0x8018,0x0001,0x8018,0x0001,
394: 0x8018,0x0001,0x8018,0x00C0,0xFF81,0x8019,0xFF81,
395: 0x8019,0x0001,0x8019,0xF001,0x8000,0x7F00,0x0000,
396: 0x0781,0x8019,0xFF81,0x8019,0xFFC0,0xC180,0x0018,
397: 0x0180,0x0010,0x0000,0x0018,0x0000,0x0000,0x0000,
398: 0x0000,0x0180,0x0018,0x0180,0x0018,0x01C0,0xC19F,
399: 0xF818,0x019F,0xF800,0x0038,0x1818,0x00E7,0xC003,
400: 0x8000,0xE000,0x39FF,0xF818,0x019F,0xF818,0x01C0,
401: 0xC180,0x1800,0x0180,0x1800,0xFFC3,0x8800,0x00E0,
402: 0x00C0,0xE030,0x380F,0x0E00,0x1800,0x0180,0x1800,
403: 0x01C0,0xC180,0x19FF,0x8180,0x190F,0xE03C,0xE1FC,
404: 0x00E0,0x03F0,0xE0FC,0x8180,0x80E0,0x19FF,0x8180,
405: 0x19FF,0x81C0,0xC000,0x1801,0x8000,0x180E,0x0038,
406: 0xE000,0xF0E0,0x038C,0x03E3,0x0C3E,0xC380,0x1801,
407: 0x8000,0x1801,0x80C0,0xDFF8,0x1801,0x9FF8,0x183E,
408: 0x3F38,0x3800,0x00E0,0x0380,0x30E0,0x0C38,0x0398,
409: 0x1801,0x9FF8,0x1801,0x9FC0,0xC018,0x0001,0x8018,
410: 0x0038,0xF8F8,0x3801,0x00E0,0x0380,0x30E0,0x0C39,
411: 0x0398,0x0001,0x8018,0x0001,0x80C0,0xC019,0xFF81,
412: 0x8019,0xFC38,0xE000,0x3981,0x80E1,0x00E0,0x3038,
413: 0x0C39,0x0398,0xFF81,0x8019,0xFF81,0x80C0,0xC018,
414: 0x0180,0x0018,0x0038,0xE000,0x3980,0x00E0,0x3000,
415: 0x3C00,0x0C38,0x0398,0x0180,0x0018,0x0180,0x00C0,
416: 0xF818,0x019F,0xF818,0x0038,0x3838,0x399F,0xC0E0,
417: 0x0C30,0xF80C,0x3C39,0x0398,0x019F,0xF818,0x019F,
418: 0xF8C0,0xD800,0x0180,0x1800,0x00CE,0xC0F8,0xF880,
419: 0x00E0,0x03F0,0xE0FC,0x8180,0x3838,0x0380,0x0180,
420: 0x0180,0x18C0,0xD9FF,0x8180,0x19FF,0x800E,0x3FE0,
421: 0xE180,0x00E1,0x8003,0xE000,0xF80E,0x0F8F,0x8180,
422: 0x19FF,0x8180,0x19C0,0xD801,0x8000,0x1801,0x8030,
423: 0xE003,0xE000,0x0038,0x800F,0x8003,0xE00E,0x0E01,
424: 0x8000,0x1801,0x8000,0x18C0,0xD801,0x9FF8,0x1801,
425: 0x9FF8,0x003F,0x8FF8,0x000E,0x00FE,0x003F,0x84C0,
426: 0x3E01,0x9FF8,0x1801,0x9FF8,0x18C0,0xC0D1,0x8018,
427: 0x0001,0x8000,0xFF8F,0x0018,0xFFFE,0x3FE0,0x0FF8,

```

Listing: Title.c

## SPEEDRUNNER (Geschwindigkeitsspiel)

Langeweile können sie nun vergessen!! Wir haben für Sie ein Spiel ■ 100 % Assembler mit über 240 Level geschrieben. Ein Leveleditor und die einfache Handhabung lassen Sie für viele Wochen nicht mehr von Ihrem Amiga los. Mehr Information hierzu in AMIGA SPACIAL 2/90 Seite 127 (Gametest).

KUNERT SOFT (SPEEDRUNNER)

nur 39,00 ■

## DANGER CASTLE (Gefährliches Schloß)

Ein unheimliches Erlebnis in einem Schloß voller Gefahren und Fallen, auch hier wurde die Motivation und Spielbarkeit mit einem saften GUT bewertet. Aber Vorsicht!!! Ihnen wird mit diesem Game ein Spiel angeboten, das wahrscheinlich Ihr Lieblingsspiel wird. Auch für dieses Spiel ist ein Spieltest in der AMIGA-Special 2/90 Seite 127 vorhanden.

KUNERT SOFT (DANGER CASTLE)

nur 39,00 DM

## MONEY PLAYER DELUXE (GELDSPIELGERÄT)

Immer noch unser Spitzenrenner im Softwarebereich. Ein Geldspielgerät mit allen Raffinessen wie Start, Stop, Risikotasten, Maus und Tastatursteuerung. Eventuelle Geldbeträge und Sonderpiele werden mit einem Zusatzmodul abgespeichert. Ihre Nerven werden beim Riskieren von Sonderspielen bis zum Zerreißen beansprucht. Auch dieses Spiel kann auf einer Festplatte installiert werden.

KUNERT SOFT (MONEY PLAYER DELUXE)

nur 39,00 DM

## KUNERT SKAT (Skatprogramm mit allen Raffinessen)

Sie werden es kaum glauben wie gut uns diese Umsetzung gelungen ist. Gespielt wird nach Original Skatregeln. Durch komfortable Menüsteuerung können Sie alles, vom Anfänger bis zum Skatprofi, einstellen. Für Anfänger sind sogar Hilfestellungen eingebaut. Endlich können Sie auf zwei Skatpartner, die meist sowieso nicht auffindbar sind, verzichten.

KUNERT SOFT (KUNERT SKAT)

nur 39,00 DM

## WIZZARD OF SOUND 2.0 (MUSIKPROGRAMM)

Wizzard of Sound ist ein sehr gutes Musikprogramm mit 61 Instrumenten, kompletter Notation, Player-Programm, Instant-Replay-Modus und einfachste Noteneingabe. Auch für Musikanfänger gut geeignet. In der Lieferung sind zwei Disketten und eine umfangreiche deutsche Anleitung auf Diskette enthalten. Eine wahre Freude für wenig Geld viel Programm (Musik). Vertriebspartner KUNERT SOFT (WIZZARD OF SOUND 2.0).

Zum sagenhaften Preis von

nur 35,00 DM

Bei Versand von Software zzgl. 6,00 DM für Porto ■ Verp. Wir suchen noch gute Programme zur Vermarktung, die den gehobenen Softwareansprüchen gerecht werden, wir bieten angemessene Bezahlung, oder eine gute Kondition.

Sie wollen sich einen AMIGA 500/2000/2500 zulegen?  
Sie haben schon einen AMIGA, aber er ist defekt?  
Sie brauchen Hardware usw. für Ihren AMIGA?  
Sie haben Probleme mit der Hardware (Drucker usw.)?

Warum fragen Sie dann nicht erst uns, wir haben für fast alle Probleme eine gute Lösung!!

## HARDWARE-ANGEBOTE

Amiga 500 Kick 1.3 Standard	898,00 DM
Amiga 2000 Kick 1.3 1 MB Chip	1898,00 DM
Amiga 2500 Kick 1.3 1 MB Chip	2298,00 DM

Laufwerk 3,5 Zoll intern A.2000	169,00 DM
Laufwerk 3,5 Zoll intern A.500	189,00 DM
Laufwerk 3,5 Zoll ex Abscha. + Bus	219,00 DM
Laufwerk 5,25 ex Absch. + Bus. + 40/80	259,00 DM

PC-AT Karte für Amiga 2000/2500	798,00 DM
PC-AT Karte für Amiga 2000/2500	1998,00 DM
Turbo board 68020 plus 2 MB 32 Bit.	2398,00 DM

Modem Discovery 1200 Baud extern	279,00 DM
Modem Discovery 2400 Baud extern	399,00 DM
Modem Supra 2400 Baud f. A. 2000 in.	389,00 DM

Festplatte 66 MB komplett fertig	1398,00 DM
Festplatte 47 MB komplett fertig	1198,00 DM
Festplatte 31 MB komplett fertig	998,00 DM

Alle 3 Platten sind einschließlich Controller, Adapter und Software für Amiga 2000/2500 voll eingerichtet, mit deutscher Einbauanleitung.  
AUTOBOOT FÜR ALLE 3 PLATTEN + 100,00 DM

**(KUNERT SOFT)**  
**COMPUTER-EXPRESS**  
**Gladbecker Straße 6**  
**4300 Essen 1**  
**Tel. 0201/312459, Fax. 0201/312469**

# Listing

```

428: 0x003F,0xF801,0x8018,0x0001,0x8018,0x00C0,0xFF81,
429: 0x8019,0xFF81,0x8010,0x0000,0x0018,0x0000,0x8001,
430: 0xE000,0x0000,0x0181,0x8019,0xFF81,0x8019,0xFFC0,
431: 0xC180,0x0018,0x0180,0x0018,0x0000,0x0018,0x0000,
432: 0x0018,0x0000,0x0000,0x0180,0x0018,0x0180,0x0018,
433: 0x01C0,0xC19F,0xF818,0x019F,0xF818,0x019F,0xF818,
434: 0x019F,0xF818,0x019F,0xF818,0x019F,0xF818,0x019F,
435: 0xF818,0x01C0,0xC180,0x1800,0x0000,0x1800,0x0180,
436: 0x1800,0x0180,0x1800,0x0000,0x0000,0x0000,0x1800,
437: 0x0000,0x1800,0x01C0,0xC180,0x19FF,0x8000,0x19FF,
438: 0x8180,0x1007,0x8180,0x19FF,0x8000,0x00FF,0x8000,
439: 0x09E0,0x0000,0x19FF,0x81C0,0xC000,0x1801,0x81C0,
440: 0x1801,0x8000,0x0071,0x8000,0x1801,0x8000,0x1C00,
441: 0x0007,0x0000,0x19C0,0x1801,0x80C0,0x0FF8,0x1801,
442: 0x8070,0x1801,0x9FF8,0x0071,0x9FF8,0x1801,0x987E,
443: 0x0700,0x07F8,0x0077,0xE1C8,0x1801,0x9FC0,0xC018,
444: 0x0001,0x8070,0x0000,0x0000,0x07F1,0x8000,0x0001,
445: 0x8071,0xE1C0,0x1F08,0x1C07,0x19C8,0x0001,0x80C0,
446: 0xC019,0xFF81,0x0671,0xC000,0x0000,0xC001,0x8000,
447: 0x0000,0x8070,0x01C0,0x1C00,0x1C07,0x07C9,0xFF81,
448: 0x80C0,0xC018,0x0018,0x061C,0x0070,0x701C,0x0070,
449: 0x0007,0x001C,0x0070,0x1870,0x7C19,0x8701,0xC018,
450: 0x0180,0x00C0,0xF818,0x019C,0xE1C,0x1860,0x1807,
451: 0x1870,0x061F,0x0077,0x0070,0x0070,0x0070,0x0760,
452: 0x1C18,0x019F,0xF8C0,0x0800,0x0180,0x1C07,0x001E,
453: 0x0787,0x0070,0x1F9C,0x0707,0x0070,0x0070,0x7000,
454: 0x0718,0x0700,0x0180,0x18C0,0x09FF,0x8180,0x7087,
455: 0x007C,0x1F07,0x0070,0x1C1C,0x7E07,0x0073,0x8070,
456: 0x71FE,0x0707,0x81C7,0x8180,0x19C0,0x0801,0x8000,
457: 0x0007,0x0070,0x1C07,0x0070,0x1C1C,0x0187,0x0071,
458: 0x8070,0x7000,0x0700,0x0671,0x8000,0x18C0,0x0801,
459: 0x9FE0,0x7F81,0xC070,0x1C07,0x0071,0x807C,0x0787,
460: 0x0070,0x01F8,0x1C00,0x1F00,0x1871,0x9FF8,0x18C0,
461: 0xC001,0x8001,0xF061,0xC070,0x1C07,0x0070,0x1FF0,
462: 0x0707,0x0070,0x01C0,0x1C00,0x1C1C,0x0071,0x8018,
463: 0x00C0,0xFF81,0x8001,0xC000,0x0707,0x1C07,0x0070,
464: 0x001C,0x0007,0x0070,0x07C9,0x8700,0x7C07,0x01F1,
465: 0x8019,0xFFC0,0xC180,0x0000,0x7000,0x0018,0x0601,
466: 0xC01C,0x0001,0xE1E1,0xC000,0x1F18,0x7807,0xF018,
467: 0x1FC0,0x0018,0x01C0,0x019F,0xF87F,0xF07F,0xFFFF,
468: 0xFFFF,0xDFFC,0x7FE1,0xD0DF,0xDFFF,0xFC18,0x07FF,
469: 0x007F,0xFCF0,0xF818,0x01C0,0xC180,0x1800,0x0100,
470: 0x0000,0x0000,0x0000,0x1C19,0xC000,0x0000,0x0000,
471: 0x0000,0x0800,0x0000,0x1800,0x01C0,0xC180,0x1980,
472: 0x0180,0x0000,0x0000,0x0000,0x0701,0xC020,0x0000,
473: 0x01FF,0x8000,0x1980,0x0180,0x19FF,0x81C0,0xC000,
474: 0x1801,0x8000,0x1801,0x8000,0x1801,0x8007,0xC801,
475: 0x8000,0x1801,0x8000,0x1801,0x8000,0x1801,0x80C0,
476: 0xDFF8,0x1801,0x9FF8,0x1801,0x9FF8,0x1800,0x7FFF,
477: 0x1801,0x9FF8,0x1801,0x9FF8,0x1801,0x9FF8,0x1801,
478: 0x9FC0,0xC018,0x0001,0x8018,0x0001,0x8018,0x0000,
479: 0x0000,0x0001,0x8018,0x0001,0x8018,0x0001,0x8018,
480: 0x0001,0x80C0,0xC019,0xFF81,0x8019,0xFF81,0x8019,
481: 0xFF81,0x8000,0xFF81,0x8019,0xFF81,0x8019,0xFF81,
482: 0x8019,0xFF81,0x80C0,0xC018,0x0180,0x0018,0x0180,
483: 0x0018,0x0180,0x0018,0x0180,0x0018,0x0180,0x0018,
484: 0x0180,0x0018,0x0180,0x00C0,0xF818,0x019F,0xF818,
485: 0x019F,0xF818,0x019F,0xF818,0x019F,0xF818,0x019F,
486: 0xF818,0x019F,0xF818,0x019F,0xF8C0,0x0800,0x0180,
487: 0x1800,0x0180,0x1800,0x0180,0x1800,0x0180,0x1800,
488: 0x0180,0x1800,0x0180,0x1800,0x0180,0x18C0,0x09FF,
489: 0x8180,0x19FF,0x8180,0x19FF,0x8180,0x19F0,0x0180,
490: 0x19FF,0x8180,0x19FF,0x8180,0x19FF,0x8180,0x19C0,
491: 0x0801,0x8000,0x1801,0x8000,0x1801,0x8000,0x1800,
492: 0x0000,0x1801,0x8000,0x1801,0x8000,0x1801,0x8000,
493: 0x18C0,0x0801,0x9FF8,0x1801,0x9FF8,0x1801,0x9FF8,
494: 0x1800,0x01F8,0x1801,0x9FF8,0x1801,0x9FF8,0x1801,
495: 0x9FF8,0x18C0,0xC001,0x8018,0x0001,0x8018,0x0001,
496: 0x8018,0x0000,0x0018,0x0001,0x8018,0x0001,0x8018,
497: 0x0001,0x8018,0x00C0,0xFF81,0x8019,0xFF81,0x8019,
498: 0xFF81,0x8019,0xFF00,0x0001,0xFF81,0x8019,0xFF81,
499: 0x8019,0xFF81,0x8019,0xFFC0,0xC180,0x0018,0x0180,
500: 0x0018,0x0180,0x0018,0x0000,0x0000,0x0180,0x0018,
501: 0x0180,0x0018,0x0180,0x0018,0x01C0,0xC19F,0xF818,
502: 0x019F,0xF818,0x019F,0xF818,0x0000,0x0000,0x019F,
503: 0xF818,0x019F,0xF818,0x019F,0xF818,0x01C0,0xC180,
504: 0x1800,0x0180,0x1800,0x0180,0x1800,0x0000,0x0000,
505: 0x0180,0x1800,0x0180,0x1800,0x0180,0x1800,0x01C0,
506: 0xC180,0x19FF,0x8180,0x19FF,0x8180,0x19FF,0x8000,
507: 0x0003,0x8180,0x19FF,0x8180,0x19FF,0x8180,0x19FF,
508: 0x81C0,0xC000,0x1801,0x8000,0x1801,0x8000,0x1801,
509: 0x8000,0x0001,0x8000,0x1801,0x8000,0x1801,0x8000,
510: 0x1801,0x80C0,0xDFF8,0x1801,0x9FF8,0x1801,0x9FF8,
511: 0x1801,0x9C00,0x0001,0x9FF8,0x1801,0x9FF8,0x1801,
512: 0x9FF8,0x1801,0x9FC0,0xC018,0x0001,0x8018,0x0001,
513: 0x8018,0x0001,0x8000,0x0000,0x8018,0x0001,0x8018,
514: 0x0001,0x8018,0x0001,0x80C0,0xC019,0xFF81,0x8019,
515: 0xFF81,0x8019,0xFF81,0x8000,0x0000,0x0019,0xFF81,
516: 0x8019,0xFF81,0x8019,0xFF81,0x80C0,0xC018,0x0180,
517: 0x0018,0x0180,0x0018,0x0180,0x0000,0x0000,0x0018,
518: 0x0180,0x0018,0x0180,0x0018,0x0180,0x00C0,0xF818,
519: 0x019F,0xF818,0x019F,0xF818,0x019F,0xE000,0x0000,
520: 0xF818,0x019F,0xF818,0x019F,0xF818,0x019F,0xF8C0,
521: 0x0800,0x0180,0x1800,0x0180,0x1800,0x0180,0x1000,
522: 0x0000,0x1800,0x0180,0x1800,0x0180,0x1800,0x0180,
523: 0x18C0,0x09FF,0x8180,0x19FF,0x8180,0x19FF,0x8180,
524: 0x1800,0x0000,0x19FF,0x8180,0x19FF,0x8180,0x19FF,
525: 0x8180,0x19C0,0x0801,0x8000,0x0001,0x8000,0x1800,
526: 0x8000,0x1801,0x8000,0x1800,0x0000,0x1801,0x8000,
527: 0x1801,0x8000,0x18C0,0x0801,0x8000,0x0001,0x9FF8,
528: 0x0000,0x0100,0x1801,0x9FF8,0x1800,0x0078,0x1801,
529: 0x9FF8,0x1801,0x9FF8,0x18C0,0xC000,0x0007,0x1C01,
530: 0x8018,0x0007,0x1C01,0xC001,0x8018,0x0000,0x0718,
531: 0x0001,0x8018,0x0001,0x8018,0x00C0,0xFF80,0x07F8,
532: 0x1C81,0x8018,0x07F8,0x1C81,0xC8F1,0x8019,0xFF81,
533: 0x8719,0xFF81,0x8019,0xFF81,0x8019,0xFFC0,0xC180,

```

Listing: Title.c

```

534: 0x1F06,0x1C80,0x0018,0x1F06,0x1C01,0xC180,0x0000,
535: 0x0180,0x0718,0x0080,0x0000,0x0000,0x0018,0x01C0,
536: 0xC190,0x1C01,0x9C9F,0xF810,0x1C01,0x9C01,0xC000,
537: 0x0000,0x0080,0x0700,0x0000,0x0000,0x0000,0x0018,
538: 0x01C0,0xC180,0x7C00,0x7C80,0x1800,0x7C80,0x7C01,
539: 0xC000,0x701C,0x1C00,0x0700,0x0700,0x701C,0x001C,
540: 0x1C00,0x01C0,0xC180,0x71F0,0x0000,0x19E0,0x7180,
541: 0x0001,0xC01E,0xE1E8,0x0707,0x8701,0xE1D8,0x707C,
542: 0xE1E8,0x073F,0x81C0,0xC000,0x7000,0x0700,0x1800,
543: 0x7000,0x0701,0xC070,0x9C07,0x861F,0x0707,0xD870,
544: 0x7670,0x7C07,0x8701,0x80C0,0xDFF0,0x7006,0x1F38,
545: 0x1800,0x73E6,0x1F01,0xC000,0x1C1F,0x061C,0x0707,
546: 0x0076,0x1870,0x701F,0x0701,0x9FC0,0xC000,0x7000,
547: 0x1C18,0x0000,0x7010,0x1C01,0xC07F,0xFC1C,0x061C,
548: 0x0707,0x0070,0x1818,0x701C,0x0701,0x80C0,0xC006,
549: 0x1C00,0x1C00,0xFF86,0x1C00,0x1C81,0xC070,0x001C,
550: 0x061C,0x0707,0x0071,0x8199,0xF01C,0x0701,0x80C0,
551: 0xC010,0x1C00,0x1C00,0x0180,0x1C00,0x1C81,0xC070,
552: 0x001C,0x061C,0x0707,0x0070,0x0181,0xC01C,0x0700,
553: 0x00C0,0xF819,0x81C0,0x7C07,0x0199,0x81C0,0x7C01,
554: 0xC81C,0x1C1C,0x0787,0x0761,0xC1F0,0x67E7,0x801C,
555: 0x071F,0xF8C0,0x8000,0x7807,0xF007,0x7006,0xC070,
556: 0xF000,0x7180,0x7807,0x01E0,0x01D8,0x07C0,0x0707,
557: 0x0007,0x01C0,0x18C0,0x09FE,0x07FF,0x007F,0x0180,
558: 0x07FF,0x007F,0xF07F,0xF7FF,0x7DFD,0xFFC7,0xFF0F,
559: 0x9F1F,0x07FF,0x7FC0,0x19C0,0x0801,0x8000,0x0800,
560: 0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,
561: 0x0001,0x8000,0x1000,0x0000,0x18C0,0x0801,0x9800,
562: 0x1800,0x9FF8,0x1800,0x9F80,0x0800,0x0800,0x0000,
563: 0x0038,0x0001,0x80E0,0x1800,0x0038,0x18C0,0xC001,
564: 0x8018,0x0001,0x8018,0x0001,0x8018,0x0001,0x8018,
565: 0x0001,0x8018,0x0001,0x8018,0x0001,0x8018,0x00C0,
566: 0xFF81,0x8019,0xFF81,0x8019,0xFF81,0x8019,0xFF81,
567: 0x8019,0xFF81,0x8019,0xFF81,0x8019,0xFF81,0x8019,
568: 0xFFC0,0xC180,0x0018,0x0801,0x0018,0x0180,0x0018,
569: 0x0180,0x0018,0x0180,0x0018,0x0180,0x0018,0x0180,
570: 0x0018,0x01C0,0xC19F,0xF818,0x019F,0xF818,0x019F,
571: 0xF818,0x019F,0xF818,0x019F,0xF818,0x019F,0xF818,
572: 0x019F,0xF818,0x01C0,0xC180,0x1800,0x0180,0x1800,
573: 0x0180,0x1800,0x0180,0x1800,0x0180,0x1800,0x0180,
574: 0x1800,0x0180,0x1800,0x01C0,0xFFFF,0xFFFF,0xFFFF,
575: 0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,
576: 0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,
577: };
578:
579: static struct Image Image1 = {
580: 0,0,
581: 234,132,
582: 2,
583: ImageData1,
584: 0x0003,0x0000,
585: NULL
586: };
587:
588: static struct Gadget Gadget1 = {
589: NULL,
590: 0,0,
591: 234,132,
592: GADGHBOX|GADGHIMAGE|GADGIMAGE,
593: RELVERIFY,
594: BOOGLADGET,
595: (APTR)&Image1,
596: NULL,
597: NULL,
598: NULL,
599: NULL,
600: NULL,
601: NULL
602: };
603:
604: #define GadgetList1 Gadget1
605:
606: static struct NewWindow NewWindowStructure1 = {
607: 217,39,
608: 234,132,
609: 0,1,
610: CLOSEWINDOW|GADGETUP|INACTIVIEWINDOW,
611: BORDERLESS|ACTIVATE|NOCAREREFRESH,
612: &Gadget1,
613: NULL,
614: NULL,
615: NULL,
616: NULL,
617: 5,5,
618: -1,-1,
619: CUSTOMSCREEN
620: };
621:
622: DisplayAbout()
623: {
624: struct IntuiMsg *msg,*GetMsg();
625: struct Window *win,*OpenWindow();
626: NewWindowStructure1.Screen=scr;
627: if (win=OpenWindow(&NewWindowStructure1)) {
628: WaitPort(win->UserPort);
629: if (msg=GetMsg(win->UserPort)) {
630: emptyqueue:
631: ReplyMsg(msg);
632: if (msg=GetMsg(win->UserPort)) goto emptyqueue;
633: CloseWindow(win);
634: return();
635: }
636: }
637: else DisplayBeep();
638: }

```

Listing: Title.c



# AMIGA-DOS-Pixel-Panorama

Was für Paris der Louvre, ist für die AMIGA DOS das Pixel-Panorama. Tatsächlich können wir Ihnen auf dieser Seite neben erlesener Raytracing-Grafik auch die ersten Bilder präsentieren, die aus den Mäusen unserer Leser geflossen sind!

**L**eben haben wir hier in der Redaktion auf Ihre Bilder warten müssen. Aber nun ist die Dürststrecke überwunden, die zwischen dem Augenblick, da diese Zeilen geschrieben werden, und dem Moment, da Sie sich zu lesen bekommen.

Peter Schütz aus der Schweiz hat uns das Jeans-Girl zugesandt. Die einzelnen Motive hat er mit Deluxe Paint III erstellt und dann mit Igi Paint zusammengefügt.

Fantasy und Science-Fiction sind die Themen, die Tobias Richter in seinen Bildern aufwirft. Er zeichnet für die fantastische Märchenstadt und das versteinerte Kampfabenteuer. Auch die Raytracing-Tisch, das Bild erklärt sich von selbst, stammt von Tobias Richter. Schließlich haben wir auch ein edles Raytracing-Bild von Richard Nicol aufgenommen. Natürlich freuen wir uns nach wie vor über Ihre illustrierten Bilder



Jeans-Girl von Peter Schütz



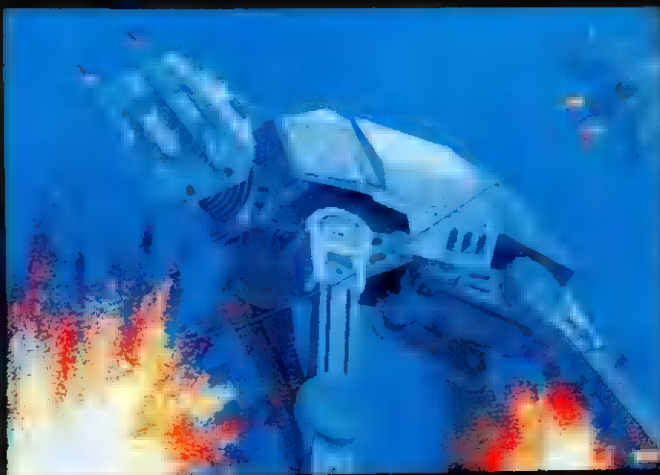
Das Land von Zamar, eine von Tobias Richters fantastischen Visionen

ganz besonders, und für jede Veröffentlichung, winkt als Dankeschön ein Software-Paket im Wert von zirka 100 DM. Mitmachen ist ganz einfach: Packen Sie Ihre Bilder auf eine Diskette, und vermerken Sie auf dem Etikett der Diskette Ihren kompletten Anschrift auch das Grafikformat der Bilder bzw. die Grafikprogramme, die bei der Erstellung verwendet wurden. Das Ganze stecken Sie dann in einen Umschlag und schicken es an den:

DDV-Vorlag  
Redaktion AMIGA DOS  
Kornwurt, Pöchl  
Postfach 250  
3440 Eschwege

Ihren Einsendungen sehen wir gespannt entgegen. Einstweilen jedoch wünschen wir Ihnen viel Spaß beim Betrachten der Bilder, die wir für diese Ausgabe ausgewählt haben, aber auch bei der Erstellung neuer.

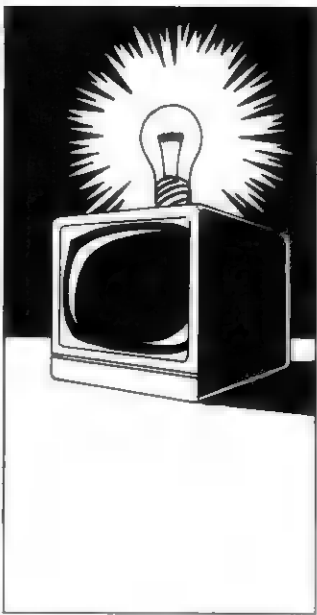
(th)



Schwarz-Ritter von Tobias Richter



Raytracing vom Feinsten, ein Bild von Richard Nicol



Edgar Meyzis

## Tips und Tricks nicht nur für M2Amiga Prozessieren & Exekutieren (Teil 1)

Über Rechtspraktiken des Staates X wollen wir weder heute noch im nächsten Heft berichten, wohl aber über Wege, aus einem Programm heraus andere mit einfachen Mitteln aufzustarten oder auch abzubrechen. Dabei muß es nicht unbedingt zu einem Prozeß kommen. Das aufzustartende Programm kann auch auf anderem Wege erzwingen, von der CPU bedient zu werden. Unsere Beispiele stützen sich auf M2Amiga/Modula 2.

Das Amiga-Betriebssystem stellt die Routine "Execute" zur Verfügung. In Modula ist die Schnittstelle zu "Dos.Execute" wie folgt deklariert: siehe Box 1.

Die Deklaration spricht für eine effiziente Implementierung. Bei Aktivierung der Compiler-Option "\$Z-" werden sämtliche Parameter direkt in Registern (d1 bis d3) übergeben. Es wird keine übliche Prozedur mit Eintritts- und Austrittscode generiert, sondern einfach Code gemäß Listing 1 anstelle des Aufrufs eingefügt [1].

### Geheimprozeß?

Die Systemroutine "Execute" entspricht nicht dem gleichlautenden CLI-Befehl. "Dos.Execute" ist dazu bestimmt, Programme zur Ausführung zu bringen, die vom CLI aus aufstartbar sind [2]. Das können sowohl die üblichen CLI-Befehle (z.B. dir, delay) als auch eigene Werke sein. "Dos.Execute" setzt eine CLI-Umgebung voraus, um stets sauber zu arbeiten. An dieser Stelle wird bereits deutlich, daß "Dos.Execute" bei der Verwendung von der Workbench aus Probleme bereiten kann. – Wir kommen in der nächsten Ausgabe darauf zurück. Die Stärke von "Dos.Execute" liegt in der Fähigkeit, auch

"BCPL-Programme" einwandfrei aufzustarten. Die originären CLI-Befehle sind in BCPL geschrieben.

"Dos.Execute" legt einen Prozeß an, ohne daß dies nach außen sichtbar wird.

### Keine Exekution ohne Prozeß

Der Prozeß trägt – wie alle mit "run" kreierte Prozesse – stets den Namen "Background CLI". Es kann mehrere "Geheimprozesse" mit diesem Namen geben. Somit ist nicht sichergestellt, daß mit der Routine "Exec.FindTask" auch wirklich die Adresse des neuen Prozesses ermittelt werden kann.

### Residentes "RUN"

"Dos.Execute" ist vom CLI-Befehl "run" abhängig, der den erwähnten Hintergrundprozeß einrichtet. Es hält sich noch immer hartnäckig das Gerücht, ein residentes "run" (im Sinne von Dos, nicht Exec) würde nicht genutzt werden. Mit der Version 1.3 des Betriebssystems besteht die Restriktion nicht mehr. Nahezu sämtliche CLI-Befehle können resident geladen werden. Es empfiehlt sich daher, die "StartUp-Sequence" um die Anweisung "resident c:run pure" zu ergänzen.

Prozesse unterscheiden sich unter anderem durch Merkmale wie Stackgröße, Priorität, das auszuführende Programm, Ein- und Ausgabekanäle, das aktuelle Verzeichnis oder das Anhängsel "CLI-Struktur". Beim Aufruf von "Dos.Execute" befinden wir zum Beispiel nicht über die Stackgröße oder die Priorität. Das Betriebssystem geht eigene Wege und verpaßt dem neuen Hintergrundprozeß die Merkmale einer bereits resident geladenen Shell oder, falls nicht vorhanden, die des im Kickstart-ROM abgelegten CLI [3]. Daraus folgt, daß zum Beispiel die Größe des Stacks nicht unmittelbar beeinflussbar ist. Ein Aufruf von "...Execute (ADR("C:STACK 40000"),...)..." würde nicht zum gewünschten Erfolg führen. Nur der von "Dos.Execute" bzw. "run" angelegte, kurzlebige (!!) Prozeß würde davon profitieren, ohne je ein anderes Programm als "STACK" auszuführen. Schade!

### Synchronisierte Prozesse

In einem Multitasking-System muß es möglich sein, Prozesse zu synchronisieren, um Abhängigkeiten Rechnung zu tragen – zum Beispiel, wenn ein Prozeß auf die Arbeitsergebnisse eines anderen angewiesen ist. Mit "Dos.Execute" steht eine perfekte "Rendezvous-Technik" zur Verfügung. Der Code, der einer "Execute-Anweisung" folgt, wird erst ausgeführt, wenn das aufgerufene Programm abgearbeitet und der

Hintergrundprozeß beendet ist. Nach außen verhält sich "Dos.Execute" somit wie ein einfacher Prozeduraufruf. Die "Rendezvous-Technik" läßt sich bei Bedarf austricksen, um asynchrone Prozesse anzulegen. Dazu ist der "commandString" mit "run" zu beginnen. Nun verhält sich "Dos.Execute" nahezu wie ein Programmaufruf mit "run" aus dem CLI. Der Programmteil hinter der "Execute"-Anweisung läuft parallel zum aufgestarteten Programm ab.

### Hintergründige Parameter

Drei Parameter sind mit Werten zu versehen. Der "command-String" (bezeichnet genau genommen die Adresse einer mit "null" (engl.), dem niedrigwertigsten ASCII-Zeichen, terminierten Zeichenkette) ist so zu behandeln, wie die Eingabezeile des CLI bei Aufstart eines Programms. Insbesondere sind auch die Umleitungen für Ein- und Ausgaben ("<", ">", ">>") zulässig. Sie werden bei der Interpretation des "command-String" erkannt und richtig umgesetzt.

Der Parameter "input" hat schon häufig falsche Erwartungen erweckt. Ein Wert ungleich "NULL" führt dazu, daß nach Abarbeitung des Programms Eingaben von der entsprechenden Datei erwartet werden. Es ist somit nicht möglich, mit "input" zu bestimmen, von welchem Datenstrom während der Ausführung (des mit Dos.Execute aufgestarteten Programms) Terminaleingaben zu lesen sind. Die Technik des Umleiten kann einspringen, sofern es ein geöffnetes "Console-Fenster" gibt und dieses dem Prozeß bekannt ist. Durch Anfügen von "<\*" an den "commandString" werden Eingaben in das "Console-Fenster" zur Verarbeitung übernommen.

Beispiel: Mit "...Execute(ADR("ReadTest <\*"),...)..." werden die vom Programm "ReadTest" geforderten Eingaben vom aktuellen "Console-Fenster" gelesen.

Ein Problem tritt hervor: Wenn kein "Console-Fenster" dem Prozeß bekannt ist, zum Beispiel bei Aufstart von der Workbench, dann stehen wir auf dem Schlauch. Wir sind

```
PROCEDURE Execute (commandString{1}:ADDRESS;  
  input{2} : FileHandlePtr;  
  output{3}: FileHandlePtr): LONGINT;  
  CODE -222;
```

Box 1. Procedure execute



somit praktisch auf eine CLI-Umgebung angewiesen, die sich durchaus schaffen läßt, wie wir in der nächsten Ausgabe sehen werden.

Betrachten wir den dritten und letzten Parameter, das heißt "output": Ein Wert von "NIL" führt dazu, daß Ausgaben im aktuellen Console-Fenster des aufrufenden Prozesses erfolgen. Bei Angabe eines Wertes ungleich "NIL" erfolgt die Ausgabe in den so bezeichneten Datenstrom, der sowohl ein "Console-Fenster" als auch eine Datei sein kann. Auch hier wird die Abhängigkeit von einer CLI-Umgebung sichtbar.

## In die Röhre geschaut

Was ist zu tun, wenn die Ausgaben eines Programms weiter zu verarbeiten sind? Ganz einfach! Wenn Ausgaben im "NIL-Device" versickern dürfen, dann müssen sie auch in eine "PIPE" geleitet werden können. Listing 2 zeigt, wie es gemacht wird. Das Programm setzt voraus, daß der "Pipe-Handler" installiert ist. Die "StartUp-Sequence" stellt sich derartigen Aufgaben ("Mount Pipe") gerne. Obwohl die "PIPE" keine Datei ist, will sie fast wie eine solche behandelt werden. Sie ist zu öffnen und zu schließen. Der "PIPE-Handler", ein Prozeß, kann mehrere "PIPES" verwalten. In unserem Beispielpro-

gramm hat die "PIPE" den Namen "P" erhalten. Beim Schreiben in die "PIPE" wird davon ausgegangen, daß nur eine Röhre offen ist.

Beim Experimentieren mit "PIPE" werden Sie schnell entdecken, daß sie ein vorzügliches Mittel zur Kommunikation zwischen Prozessen abgibt [4]. Sie werden auch bestätigt finden, daß "PIPES" mit "oldFile" zu öffnen sind, um uneingeschränkt nutzbar zu sein. Es werden aber auch die Grenzen von "PIPES" deutlich: Sie sind nur vier kByte "lang" und verhindern praktisch den Dialog mit dem aufgestarteten Programm. Das mangelnde "Fassungsvermögen" wirkt sich jedoch nur bei synchron ablaufenden Prozessen aus, wenn zwischendurch nicht "abgepumpt" werden kann.

## Trügerisches Ergebnis

Die Systemroutine "Dos.Execute" liefert als Ergebnis null (DosFALSE) oder minus eins (DosTRUE). Mit DosFalse wird signalisiert, daß "run" zum Anlegen des Hintergrundprozesses nicht verfügbar war, weder in der von Dos geführten Liste der resident gehaltenen Programme noch im Verzeichnis "C". Folglich wurde auch nicht der Versuch unternommen, das mit Parameter eins bestimmte Programm aufzustarten.

Das Ergebnis DosTRUE bedeu-

tet aber noch lange nicht, daß das gewünschte Programm aufgestartet werden konnte. Es ist somit nicht möglich, aus dem Ergebnis auf das Gelingen der gesamten Operation "Execute..." zu schließen. Als Programmierer übernimmt man die Verpflichtung, dafür zu sorgen, daß das aufzustartende Programm vorhanden ist und mit dem richtigen Pfad in den "commandString" eingetragen wurde. Kann man Pfad und Namen nicht schon bei der Programmierung fest eintragen, so bleibt zur Laufzeit nichts anderes übrig, als die Liste der resident geladenen Befehle zu durchsuchen, oder sich mit "Dos.Lock/Dos.Unlock" von der Existenz der Programmdatei zu überzeugen.

Über Dateioperationen des aufgestarteten Programms sind jedoch Informationen nach Auswertung des Sekundärergebnisses mit der Systemroutine "Dos.LoErr()" verfügbar. Es werden die bekannten Dos-Fehlercodes gemeldet. Mit Null wird signalisiert, daß kein Fehler auftrat.

Listing 3 zeigt, wie bei der Auswertung der beiden möglichen Ergebnisse vorgegangen werden kann. Falls "run" zur Erzeugung der Hintergrundprozesse nicht ladbar ist, so wird der selbstdefinierte Fehlercode "55" gemeldet. Sollte das Verzeichnis "X" nicht vorhanden sein, dann ermittelt "Dos.LoErr()" den Fehlercode "205" (object not

found). Beachten Sie bitte, daß ein explizit aufgeführtes "run" zur Bildung eines asynchron ablaufenden Prozesses durch keinen Fehlercode erfaßt wird, dafür aber die Bemühung, das so aufzustartende Programm zu laden. Ein Fehlercode ungleich null kann aber auch durch sich anschließende Dateioperationen hervorgerufen sein.

## Ausblick

In der nächsten Ausgabe werden wir die heute erworbene Erfahrung in einer CLI-Umgebung auf die Workbench übertragen und Wege aufzeigen, die Notbremse zu ziehen, das heißt, asynchrone Prozesse abubrechen. Weiterhin werden wir synchrone Prozesstechniken ersetzen und asynchrone Prozesse "von Hand schmieden".

(mm)

### Literatur:

- [1] M2Amiga Benutzerhandbuch, 1989
- [2] The Amiga-Dos-Manual, 1989
- [3] Das Amiga Guru-Buch, 1989
- [4] Amiga ENHANCER SOFTWARE Version 1.3 bzw. neuere Begleitdokumentation zum Amiga (Amiga-Dos)

## Listings

```
1:  lea    -86(pc),a3 ; Adresse auf commandString
2:  move.l a3,d1      ; in Parameter 1
3:  moveq  #0,d2      ; input
4:  moveq  #0,d3      ; output
5:  move.l -66(pc),a6 ; Basisadr. Dos-Lib.
6:  jar    -222(a6)    ; Dos.Execute ausführen
7:
8:
```

Listing 1: Aufruf der Systemroutine "Dos.Execute" ohne Umweg über den Stack

```
1: MODULE Pipe;
2:
3: FROM Dos IMPORT      Execute, IoErr;
4: FROM SYSTEM IMPORT  ADR;
5:
6: CONST DosTRUE = -1;
7:       DosFALSE = 0;
8:       NoRun    = 55; (* privater Code *)
9:
10: VAR ergebnis,
11:     fehler : LONGINT;
12: (* $Z- *)
13: BEGIN
14:   ergebnis := Execute (ADR("dir x"), NIL, NIL);
15:   IF ergebnis = DosTRUE THEN (* Prozess angelegt *)
16:     fehler := IoErr() (* Konnte_dir arbeiten ? *)
17:   ELSE
18:     fehler := NoRun (* Prozess nicht angelegt *)
19:   END;
20: END NO.
```

```
21:  Close (fileHdl);
22:
23:  (* und nun die PIPE: leeren *)
24:  ergebnis := Execute (ADR("c:type PIPE:P"), NIL, NIL);
25:
26:  END;
27: END Pipe.
```

Listing 2: PIPES sind ein hervorragendes Mittel zur Kommunikation zwischen Prozessen

```
1: MODULE X0;
2:
3: FROM Dos IMPORT      Execute, IoErr;
4: FROM SYSTEM IMPORT  ADR;
5:
6: CONST DosTRUE = -1;
7:       DosFALSE = 0;
8:       NoRun    = 55; (* privater Code *)
9:
10: VAR ergebnis,
11:     fehler : LONGINT;
12: (* $Z- *)
13: BEGIN
14:   ergebnis := Execute (ADR("dir x"), NIL, NIL);
15:   IF ergebnis = DosTRUE THEN (* Prozess angelegt *)
16:     fehler := IoErr() (* Konnte_dir arbeiten ? *)
17:   ELSE
18:     fehler := NoRun (* Prozess nicht angelegt *)
19:   END;
20: END NO.
```

Listing 3: DosTRUE und IoErr() = 0 als Mindestvoraussetzung, daß alles glatt ging

**R**alf Leithaus, der schon als Mitherausgeber der Public-Domain-Bücher aus dem Technik-Support-Verlag verantwortlich zeichnet, hat – diesmal in Eigenregie – ein Lehrbuch für den Umgang mit dem AmigaDOS herausgebracht. AmigaDOS für Anwender, so der Titel, ist aber nicht nur für Fortgeschrittene gedacht, nein, auch der Neuling findet einen gelungenen Einstieg in den Umgang mit den DOS-Versionen 1.2 und 1.3.

## AmigaDOS für Anwender

Nach jedem Kapitel wird der gesamte Inhalt noch einmal kurz zusammengefaßt und der gelernte Stoff durch verschiedene Aufgaben abgefragt. Die Lösungen zu den einzelnen Aufgaben sind im Anhang des Buches zu finden. Diese Art der Lesereinbeziehung findet besonderen Anreiz und lockert den doch stellenweise recht trockenen Lehrstoff des Betriebssystems etwas auf. Somit kann der Leser aktiv beim Durcharbeiten des Stoffes mitarbeiten. Nachdem grundsätzliche Begriffe geklärt wurden, werden anhand von einfachen Beispielen dem Leser einige Kopierkniffe beigebracht. Hier findet man auch vieles, was im Handbuch einfach nicht enthalten ist. Nachdem der Unterschied zwischen dem CLI und der Shell herausgestellt wurde, ist der Leser schon im nächsten Kapitel. Auch hier wird mit diversen Beispielen der Umgang mit den Directories und Subdirectories veranschaulicht. Aber auch die logischen Devices, ein Problem, mit dem sicherlich jeder Amiganer schon einmal zu kämpfen hatte, werden schnell plausibel gemacht. Warum kann man so einfache Befehle wie DIR oder COPY überhaupt ausführen? Fragen, über die oftmals nur gelächelt, aber dennoch keine Antwort gefunden wird, werden hier klipp und klar beantwortet. Dieses Basiswissen versucht das Buch dem Leser zu vermitteln, damit er die gesamte Funktionsweise des AmigaDOS versteht. Auch das Prinzip des Multitasking versucht der Autor mit einigen Beispielen dem Leser näherzubringen. Fer-

ner werden die verschiedenen Dateitypen und deren Funktionen hervorgehoben. In einem weiteren Kapitel werden dann Peripheriegeräte, wie Diskettenlaufwerke, Modem, Drucker usw., vorgestellt. Das letzte Kapitel beschäftigt sich mit der Festplattenverwaltung. Hier wird die Installation und das Ansprechen bzw. die vollständige Nut-

Bastlerkreis will das Buch Amiga-Hardware-Tuning ansprechen.

## Amiga-Hardware-Tuning

Aber auch an diejenigen, die die Funktion der Hardware ihres Computers besser verstehen möchten, will sich dieses Buch richten. Dabei besticht das Buch auf den ersten

sächlich um das sagenumwobene Write-Only-Memory, kurz WOM genannt?), ein Analog-Digital-Wandler, der eigentlich als Bilderzeuger im Amiga ein Digital-Analog-Wandler ist, bis hin zum fatalen Fehler im Bestückungsplan des Epromers. Dort ist das IC 74LS86 falsch gepolt, das heißt, die alles entscheidende Kerbe müßte eigentlich auf die andere Seite zeigen. Dieses kleine Mißgeschick hat zur Folge, daß sich der Epromer beim ersten Test sofort mit einer kleinen Rauchwolke in die ewigen Jagdgründe verabschiedet. Sollte man es dennoch gemerkt und das IC richtig bestückt haben, so hat man bestimmt den unerlaubten Kontakt zweier Leiterbahnen unter dem Eprom-Sockel übersehen, der bereits als Fehler auf dem Platinen-Lay-out anzutreffen ist.

Trotzdem uns diese Fehler gerade noch rechtzeitig auffielen, war es uns nicht möglich, den Epromer in Betrieb zu nehmen, da die auf der Diskette mitgelieferte Software völlig versagte: Bereits sofort nach Aufruf des Brenners verabschiedete sie sich ohne Anzeichen eines Fehlers und ward zur Speicherleiche. Ganz ähnlich verhielt sich das Programm, das die Dateien zum Brennen der Eproms für die ROM-Disk erstellen sollte. Obwohl nur sechs Eproms vorgesehen waren, erstellte es neben den Dateien ROM1 bis ROM6 auch noch ROM129 und ROM130, wenn es nicht sogar mit einem 'Software Error - Task held...' den Guru meditieren ließ.

Den Test weiterer Soft- und Hardware haben wir unseren Computern erspart, da wir der Fehler überdrüssig waren und kein weiteres Risiko eingehen wollten.



zung einer Festplatte deutlich hervorgehoben. Neben den schon erwähnten Lösungen zu den Aufgaben werden im Anhang zusätzlich die Fehlercodes des AmigaDOS aufgelistet.

(Jürgen Seibel/br)

Name: AmigaDOS für Anwender  
Autor: Ralf Leithaus  
Verlag: Technik Support  
ISBN: 3-926847-09-3  
Preis: 49,- DM

**E**in bißchen Tuning kann nicht schaden. Das sagen sicherlich nicht nur diverse Autofahrer, sondern auch viele Amiga-Besitzer. Und genau diesen

Blick durch die gelungene Auswahl der Erweiterungen, mit denen man im Selbstbau seine "Freundin" aufrüsten kann. So werden beispielsweise eine RAM/ROM-Disk, deren Platine dem Buch beiliegt, ein Epromer, ein MIDI-Interface, ein Genlock-Interface sowie ein Sampler zum Selbstbau besprochen. Da schlägt das Bastlerherz höher. Doch die anfängliche Freude ist schnell vorbei. Spätestens wenn der Amiga kaputt in der Ecke steht, fragt man sich, was einem die Anschaffung dieses Buches eigentlich Positives gebracht hat.

Angefangen bei kleinen Fehlern wie eine ROM-Disk, die den Status 'Write-Only' trägt (oder handelt es sich hier tat-

Name: Amiga-Hardware-Tuning  
Autoren: Uwe Gerlach und Christian Hochberger  
Verlag: Markt & Technik  
ISBN-Nr.: 3-89090-586-2  
Preis: 98,- DM

Um alle Erfahrungen auf einen Nenner zu bringen, könnte man sagen: Gut gedacht, schlecht gemacht. Selbst einem Profi auf diesem Gebiet würde die Inbetriebnahme der Erweiterungen nur mühsam oder überhaupt nicht gelingen. Von den Problemen eines Amateurs ganz zu



schweigen. Deshalb unsere Meinung: Wer für dieses Buch 98,- DM bezahlt, ist selber schuld.

(Bernd Rudolf/br)

**D**er Amiga – das unbekannte Wesen. Viele Amiga-Besitzer stolpern immer dann, wenn Sie tiefer in die Programmierung des Computers einsteigen wollen, über Verständnisschwierigkeiten, die vor allem das Betriebssystem und den internen Aufbau des Amiga angehen. Die meisten auf dem Markt befindlichen Bücher sind einfach zu speziell auf Themen ausgerichtet; ein Nachschlagewerk zum 'Eben-mal-Reinschauen', das die fundamentalsten Fragen beantwortet, befindet sich jedoch kaum darunter. Was bleibt also? Für jedes Programm, das man erstellen will, muß eine Masse Literatur platztilgend neben dem Rechner liegen. Der Sybex-Verlag hat sich dieser Thematik angenommen und durch seine Autoren Garry Glendown und Roland Haas ein komplettes Nachschlagewerk erstellen lassen – das Amiga Profi-Buch.

## Das Amiga Profi-Buch

Dieses Werk richtet sich vor allem an die Insider des Bereichs Amiga; dementsprechend wurde auch wenig auf die Grundsätze zur Hard- und Software eingegangen. Trotzdem zielt das Buch auf einen festen Bestandteil vom Equipment des Nicht-nur-Anwenders.

Was bietet das Profi-Buch dem Leser? Zunächst einmal komplexe Information zur Hardware des Amiga. Seine Besonderheiten, die Customchips, die Portbausteine, DMA-Technik, all das, was die Technik des Amiga ausmacht, wird unter Zuhilfenahme anschaulicher Grafiken verständlich gemacht. Dabei legen die Autoren auch Wert auf die Tatsache, daß es unterschiedliche Versionen des Amiga gibt. Soll heißen, daß hier keine Spezialisierung auf die einzelnen Typen stattfindet, sondern auch Unterschiede nebeneinander aufgezählt werden.

Als nächstes wird die CPU (der Motorola 68000) untersucht. Blockdiagramme und Grafiken über enthaltene Register bieten dem Leser die Möglichkeit, nach Lesen des Textes das Erlernte auch anhand eines Schaubildes nachzuvollziehen. Die Pin-Belegung des Prozessors sowie die Erklärung der einzelnen Signale, die an den Prozessorspins anzuliegen haben, fehlen im Buch auch nicht. Weiterhin werden dem Leser die Unterschiede des 68000 zu seinen 'Nachfolgern' 68010, 68020 und 68030 erklärt, so daß sich jeder ausmalen kann, wo bei einem eventuellen Wechsel des Prozessors die Stärken und Schwächen am meisten zum Tragen kommen.

Die Customchips werden dazu nicht nur einfach erwähnt, sondern ihr interner Aufbau ebenso präzise behandelt wie der des Hauptprozessors, dito die CIA-Bausteine. Wer schon immer etwas über die enthaltenen Timer oder die Programmierung der Ports wissen wollte, ist beim Profi-Buch genau richtig. Serieller und paralleler Port werden komplett erklärt, der Aufbau, die Pinbelegung und die anliegenden Signale sind genauestens beschrieben und trotzdem nicht nur für Techniker verständlich. Gleiches trifft auch für die Kapitel über den direkten Speicherzugriff und über die Tastatur-Decodierung zu.

Im allgemeinen bekommt man bei den Hardwarebeschreibungen stark das Gefühl, daß hier endlich einmal Wert auf Wissensübermittlung und nicht nur auf Wissensdarstellung gelegt wurde. Die Hardware bestimmt zwar einen großen Teil des Gesamtwerkes, die Systemsoftware jedoch bildet eindeutig den Schwerpunkt. Hier merkt man, daß das Buch an Fortgeschrittene und Profis gerichtet ist: Fast alle Programmbeispiele sind in der Sprache 'C' geschrieben, viele Erklärungen sind außerdem mit Assembler-routinen unterlegt. Kein Wunder, da sich mit diesen beiden Sprachen am effektivsten das Amiga-System einsehen läßt. Angefangen wird mit der Exec-Base, dem Grundstein zum Arbeiten mit dem Betriebssystem. Die einzelnen Vektoren der Exec-Base werden beschrieben und ihre Funktionen ausführlich dargestellt. Über die Tasks,

Interrupts, Messages, Ports und Speicherzugriffe geht man langsam aber stetig in Richtung Intuition-Programmierung, einem der interessantesten, aber auch schwersten Kapitel zum Thema Amiga-Programmierung. Aber auch hier merkt der Leser schnell, daß die Autoren nicht zu den oberflächlichen Schreibern gehören, sondern ihr gesamtes Wissen dem Leser zur Verfügung stellen.

Die beiden nächsten Kapitel befassen sich mit dem sogenannten "Disk-Operating-System" des Amiga. Auch hier wird wieder ins Detail gegangen: Der Aufbau der Disketten wird anhand der Beschreibung des Root-Blocks, des Directory-, File-Header-, File-List- und Data-Blocks gezeigt. Im Anschluß daran, werden alle Library-Routinen erwähnt, dazu passend Deklarationen und Aufrufe in 'C' und Assembler mit zusätzlicher Beschreibung der benötigten Parameter. Und wem bis jetzt das Buch noch nicht geholfen hat, der findet spätestens hier eine wertvolle Unterstützung beim Programmieren.

Im Anhang finden sich noch die Befehlsausführungszeiten des Prozessors, eine alphabetische Auflistung aller Customchip-Register, die Guru-Meditation-Fehlermeldungen, die Library-Routinen nach Offset sortiert, eine Liste der Tastencodes, Beschreibung des IFF-Formates, C-Operatoren und ihre Prioritätsverteilung sowie eine Beschreibung der Programme, die auf der dem Buch beiliegenden Diskette zu finden sind.

Dieses Buch kann auf jeden Fall denen empfohlen werden, die konsequent die Programmierung des Amiga mittels systemnaher Sprachen fortführen oder ihr Wissen über die internen Vorgänge des Computers erweitern wollen.

(jb/br)

Name: Amiga Profi-Buch  
Autor: Glendown, Haas  
Verlag: Sybex  
ISBN: 3-88745-580-0  
Preis: 79,- DM

lassen sich im DOS umständliche Operationen immens vereinfachen und um ein Vielfaches beschleunigen. Das Buch 'Die besten Amiga Utilities' soll nun dem kaufwilligen Benutzer eine Orientierung im Dschungel der Utilities geben. Zugleich sollen die Amiga-Besitzer, die zugleich die im Buch vorgestellten Utilities besitzen, die gesamte Kapazität dieser Tools nutzen können, da manchen Programmen meist nur käglige Beschreibungen beiliegen oder die Handbücher nur in englischer Sprache gehalten sind.

## Die besten Amiga Utilities

In dem Buch werden 21 Dienstprogramme vorgestellt, die das Arbeiten mit dem Amiga erleichtern sollen.

Jedes Utility wird mit einem Vorwort eingeleitet, das die Funktion des jeweiligen Programms vorstellt. Im folgenden werden die einzelnen Menüs und Befehle gezeigt. Kleinere Anwendungsbeispiele sollen dem Leser den Umgang mit seinem Programm dabei einfacher gestalten.

Im Anhang findet der Leser schließlich die Bezugsadressen der vorgestellten Utilities. Hier eine Zusammenstellung der Programme, die im Buch erwähnt werden:

Autoboot-Ramdisk  
Butcher V2.0  
Create-A-Shape  
R.C.T  
Aztec 3.6a  
Boot-Selector,  
PowerWindows 2.5  
CygnusEd Professional  
Sprite Animator  
Hires-Workbench  
WShell  
Discovery  
Diskmaster  
Dos-2-Dos  
Sherlock  
Modeler 3D  
B.A.D  
Quaterback  
TurboPrint II  
Zenon  
Zing!Keys

(Jürgen Seibel/br)

Name: Die besten Amiga Utilities  
Autor: Andreas Polk  
Verlag: DATA Becker  
ISBN: 3-89011-108-4  
Preis: 39,- DM

### Holt Euch zu AMIGAtionellen Preisen!!



**Staubschutzhauben** DM 39,-  
im New Art Design

**Speichererweiterung A500** DM 189,-  
intern, abschaltbar mit Uhr

**extern. Laufwerk** DM 199,-  
mit durchgeführtem Bus, abschaltbar

**Citizen Swift 24** DM 998,-  
der Testsieger unter den 24 Nadel-Farbdruckern

bei: Miky Wenngatz  
Jägerweg 31, 8031 Gilching  
Telefon 081 05/24540

Fordert unseren kostenlosen Katalog an

### Commodore® Ersatzteil Service

Wir liefern  
für **Händler** und Privat-  
anwender preiswert und prompt

Rufen Sie uns an: (02331-43001)  
oder schreiben Sie uns:

CIK-Computertechnik ■ Ingo Klepsch  
Berliner Straße 49b ■ D-5800 Hagen 7

**TELEFAX: 02331-42499**

### UNGLAUBLICH !!

Aber wir haben wirklich über  
7000 AMIGA-PUBLIC-DOMAIN !!!

kopiert auf 3,5" .. ab 2,60  
kopiert auf 5,25" .. ab 1,40

5 deutsche Katalog-Disk (3,5") +  
die neuste TIME : DM 20,- (VK)

diverse Super- Sonder-PD :  
z.B. Return to earth Vers. 1.1 : 5,-  
AVD mit VirusX 4.0 u.v.m.: 5,-

Info-Disk (3,5") DM 1,-

A.P.S. -electronic- BCom Datentechnik

Sonnenborstel 31 Chemnitzer Str. 82  
D-3071 Steimbke D-3320 Salzgitter  
Tel.: 05026/1700 Tel.: 05341/46954  
FAX 05026/1615 FAX 05341/15061

**10 000** AMIGA  
MS-DOS  
Public Domain kopiert auf

**2DD Markendisketten**  
z.B. Maxell, TDK, C=

ab **2,90** No Name  
ab 2,- DM

Amiga: 5 Katalog-Disk, 10,- DM in  
Briefmarken, MS-DOS Katalog 5,- DM

**Selma Fester**

A. d. Alpheide 26B, 3070 Nienburg  
Telefon 0 50 21/6 49 25

### AMIGA PD

Riesiges Angebot ausgesuchter PD Software

- Jedes Programm mit DEUTSCHER ANLEITUNG
- SUPER SICHER - Alles ■ Virenfrei

**zum Sonderpreis 1,-**

Immer auf dem NEUESTEN STAND

- Zu jeder AMIGA-AUSGABE ■ getesteten  
PD PROGRAMME auf DREI 3,5" Disketten  
in einem Monatlichen ■ zusammengefaßt für:

DM 19.90 per Bankinzug ■ ■ per Nachnahme

**Jeder einmalig für Sie 9,90**  
**Schnippelpreis**

- Fordern Sie unseren umfangreichen Katalog  
für ■ ■ 5,- ■

Hotline 0781-57734

von 10.00 - 20.00 Uhr

GERMAN - SOFT

PD - UTE - URLBAUER

424 C-64/C-128/AMIGA PD

Am Rönning 8 D 7800 Osttignau 18

tel. 0781 57734

### Unverschämt preiswert

Disketten sofort lieferbar - Qualitätsdisketten  
100 % Umtauschgarantie !

3,5" 2DD/10 St. - 12,50 Tagespreise !

Fuji Disketten ■ 10 St. ■ 100 St.

3,5" 2DD/10 St. 27,- 26,-

3,5" 2DD pink, grün 28,- 27,-

Speichererweiterungen andere  
auf Anfr.

■ 500, 512 KB, abschaltbar 159,-

A 1000, 2 MB Ram Box, 512 KB best. 444,-

■ 2000, 8 MB, ■ MB best. 844,-

Drucker: Star LC 24-10 679,-

Star LC 10 Color 589,-

andere Modelle auf Anfrage!

Kostenlose Preisliste anfordern, auch für Ausland !

AFM Computer

Fachweg 2010, 7886 Mügg. Tel. 07143 1234

### Amiga Public Domain

24 h Expressversand

ACS	- 229	AMIGAWORLDOOLCHEST	- 5
ANTARES	- 39	■ ■ ■ 4000	- 41
AUSTRIA	- 8	BARRACUDA	- 11
CACTUS	- 35	CHIRON	- 136
DBW RENDER	- 7	DDO	- 13
EROTIC	- 147	FAUG	- 85
FRED FISH	- 310	FOUNTS	- 11
FRANZ	- 53	GET IT	- 206
HIGHLIGHTS	- 33	ICONS	- 8
KICKSTART	- 250	KISS	- 134
NICLAS	- 11	OASE	- 49
PANORAMA	- 324	PE N I S	- 3
PUBLIC-PROJECT	- 5	RAY TRACING	- 7
RPD	- 200	RUHR	- 28
RW	- 17	S.A.F.E	- 36a
TAIFUN	- 120	TBAG	- 32
TORNADO	- 30		

\* nur ■ ■ ■ Altersnachweis - Alle Serien ständig aktualisiert

50 Stück 3,5" 2DD Qualitätsdisk **DM 80,-**

Einzeldisk	DM 4,50	Qualitätsgarantie
■ 10 Disk	DM 4,00	Wir kopieren nur mit
ab 50 Disk	DM 3,50	doppeltem verify - ■ ■ ■
■ 100 Disk	DM 3,00	virus-u. errorfrei. - Etikettiert

Hardware: 512 KB Speichererweiterung DM 199,-

■ Katalogdisk DM 5,- (V-Scheck/Briefmarken)

Zzgl. DM 4,- bei Vorkasse, bei Nachnahme DM 6,-

Ab ■ ■ Disk bei Vorkasse versand-

kostenfrei.

**ALPHA-SOFT**

E. Schlick

Postfach 105 - 6719 Cottberg

Hotline: 0 53 56-52 84

## Ihre Hotline für AMIGA-DOS-TIP

Tel.: 089/4391087 bis -89

Fax: 089/4391080



### Mit Aktien Geld verdienen durch **AMIGA-Börse '90** Das Aktienprogramm für den Profi wie für den Einsteiger

★ Mit den Features vielfach teurerer Programme ★ Zum Beispiel:  
★ Konkrete Kaufsignale und Verkaufssignale ★ Aktienzahlen beliebig  
★ Charts ★ Gleitende Durchschnitte ★ Kursverfolgung über 4 Jahre  
★ Ranglistenwerte sortierbar ★ Technische Aktienanalyse ★ Druckoption  
★ Auswahl internat. Aktien auf der Diskette ★ vieles andere mehr...  
Dazu ■■■■ Benutzeroberfläche vom Feinsten, Pull-down-Menüs, Funktionsfelder, Funktionstastenbelegung, Mausbedienung, Lauffähig auf allen Amigas ab 512 KByte und von der Festplatte!

Programmdiskette und Handbuch ■■■■ 178,-  
DEMO ■■■■ 30,-

zzgl. Versandkosten im Inland inkl. ■■■■ 12,-, Ausland DM 15,-  
Wir liefern innerhalb der BRD gegen Nachnahme, ins Ausland gegen Vorkasse (Euroscheck).

**GUSSENBAUER**  
Panoramastr. 18 · 7107 Nordheim  
Tel.: 0 71 33/49 25

### Speichererweiterungen

ANRUFEN!!!! LOHNT SICH!!!!  
TAGESPREISE!!!!

Amiga 500	- 512 ■■■■ intern, abschaltbar	ab DM	159,-
Amiga 500	- 1 MB intern, absch., Uhr	ab DM	398,-
Amiga 500	- 2 MB intern, absch., Uhr	■ ■ ■ ■	598,-
Amiga 1000	- 2 MB extern, absch.,	■ ■ ■ ■	DM 798,-
Amiga 2000	- ■ MB intern, absch.,	ab DM	898,-

Alle Speichererweiterungen sind autokonfigurierend abschaltbar und mit sehr schnellen RAM's (100ns und schneller) ausgerüstet!  
Durch Megabit-Technologie: minimaler Strombedarf  
\*\*\*\*12 Monate Garantie\*\*\*\*

Laufwerk 3.5" intern f. Amiga ■■■■	DM	169,-
Laufwerk 3.5" ■■■■, durchschaltbar ■■■■ abschaltb. DM	198,-	

**B & S Computer-Vertrieb GmbH**

Beethovenstr. 33 · 4172 Straelen  
Tel: 02834/1249 · Fax: 02834/6979

**MacSoft** - AMIGA SHOP  
Hardware - Software - Schulung - PD

### Public Domain DISKETTE AUF 2 DD NUR **4,- DM**

Über 4500 PD-Disk! Immer aktuell!  
24-Std.-Versand-Service  
Katalog-Disketten anfordern, 5,- DM  
Selber abholen, NN gespart!  
Hardware-Zusammenstellung auf Wunsch.  
Lassen Sie sich ihren persönlichen Amiga anfertigen.  
Fragen Sie nach unseren Amiga-Einsteiger-Kursen.

Telefon 02 31 / 51 60 10

Mo.-Fr. 10-13, 15-20 Uhr, Sa. 10-16 Uhr

Hannörrischestr. 82, 4600 Dortmund 1  
BTX \* mac soft amiga #

### COMPUTER SHOP

WIE?... NEU! NEU! NEU! NEU! NEU!  
AMIGA Schweiz AMIGA Schweiz

WAS?... AMIGA und alles für den  
AMIGA bis Professional ...

WO?... P.V. COMPUTER-SHOP  
Bubikon ZH, Ladengeschäft  
Neu! in 8623 Wetzikon Neu!  
an der Bahnhofstr. 278  
Tel. 01-9307954

HARDWARE • SOFTWARE • BERATUNG

Von 9.30 bis 22.00 UHR  
**AMIGA PD SO GÜNSTIG**  
**WIE NOCH NIEMALS ZUFÜR**  
**WIR KOPIEREN MIT VERLIER**  
**ÜBER 4500 PD-DISK**

R. Dombrowski 040/ 642 82 25  
Postfach 71 04 62  
2000 Hamburg 71  
NEU 24 Std. Versand-Service ohne Aufpreis.

3.5" 2DD	5.25" 2D
PD incl. Qualitätsdisk	PD incl. Qualitätsdisk
1 - 9 a DM 2,80	10 - 39 a DM 1,30
10 - 79 a DM 2,00	40 - 99 a DM 1,10
80 - a DM 1,90	100 - a DM 1,00
Serienabnahme ab 200 PD a 1,70 DM	Serienabnahme ab 300 PD a 0,95 DM

WIR HABEN 80 SERIEN ALLE UNSERE SERIEN SIND IMMER AKTUELL 60!  
**ABO Möglichkeit!** EIGENE "ANTARES" Disk  
5.25" Markendisk Zuschlag je Staffeln von 0.40 DM a Disk  
5.25" Farbdisketten Zuschlag je Staffeln von 0.20 DM a Disk  
**PD incl. 3.5" 2DD Markendisk.**  
1 - 9 a DM 3,00 10 - 79 a DM 2,40  
80 - a DM 2,30 **Sensationelle Neuheiten**

ABSOLUT NEU: deutsche Katalogdiskette: auf dieser Diskette ist der ■■■■ von ■■■■ normalen Katalogdisketten enthalten incl. ANTARES Menü 4.-DM incl. Portonur bei Vorkasse 1 Briefm! Nachnahme Vorkasse (nur Scheck oder Überweisung kein Bargeld + Porto 8.00DM Nachnahme 8.00DM incl. Verpackung

### IMMER UP-TO-DATE AMIGA PD Wir führen (fast) alle Serien 24 Std. Versand

■ ■ ■ ■	ab 2,68 DM
■ ■ ■ ■	ab 1,68 DM

Verify-Etikettiert-auf Viren  
geprüft  
**OTP - PAKET**

10 Disketten mit ca. 200 Fonts und 200 Clips für  
Pagesetter, Deluxe Paint usw. incl. 80 Druckertreiber  
Mit ausführlicher Dokumentation und Illustration

D. Ptak A. Fuchs  
St. Ingbert Trier  
06894/381331

**Musik- und Grafiksoftware Shop**  
Wasserburger 244 \* 8000 München 82  
Telefon: 089 62 62 62

### "THE QUEST SEQUENZER"

Das neue 24-Spur Sequenzerprogramm für alle AMIGA.

Das bekannte Sequenzerprogramm "TEXTURE" ist bereits seit 1985 eines der erfolgreichsten Sequenzerprogramme auf dem ■■■■. Endlich mit dem Programmator Roger Powell und Sound Quest die Umsetzung für den Amiga gelungen. TEXTURE wurde durch ■■■■ bekannte Anwender wie Jan Hammer und Stevie Wonder bekannt.

Die Bedienung erfolgt entweder über die Tastatur oder direkt mit der AMIGA-Maus. Das Programm bedient sich einer ausgefeilten PULL-DOWN-MENU-Technik, um eine optimale Bedienungsführung zu gewährleisten. Dabei wurde vor allem Wert ■■■■ optimales Timing gelegt, sodaß der AMIGA nun auch studiotauglich geworden ist. Zahlreiche Funktionen erleichtern das Aufnehmen, Arrangieren und Manipulieren von Midi-Daten. Alle Funktionen können in Realtime während des Abspielens ohne Timingprobleme aktiviert werden. Der QUEST SEQUENZER läuft auf allen AMIGA Modellen ab 512 KByte RAM und mit allen Standard-MIDI-Interfaces.

Preis: nur DM 298,-

Außerdem führen wir Editoren für viele gängige Synthesizer von Roland, Yamaha, Casio, Ensoniq u.a.

Kostenlos: AMIGA-MIDI-Konverter! (Rückporto)

Rufen Sie uns einfach an oder besuchen Sie uns in unserem Laden

Daten- und Organisationssysteme  
Hard- und Softwarevertrieb

Ihr AMIGA-Fachhändler  
im Bergischen Land!



### Filecard A-2000 AutoBoot

21 MB 40ms MFM.....999 ■■■■	47 MB 28 ms RLL.....1.499 ■■■■
32 MB 40ms RLL.....1.099 DM	88 MB 15 ms MFM.....2.849 DM
47 MB 40ms RLL.....1.459 DM	133 ■■■■ 15ms RLL.....3.399 ■■■■

Alle Cards inkl. ALF 2.0 Software, AutoPark, AutoBoot unter Kick 1.3. Einbau im Preis.

### HardDisk A-500/1000 AutoBoot

20 ■■■■ 40ms MFM.....1.149 DM	48 ■■■■ 28ms RLL.....1.649 DM
32 ■■■■ 40ms RLL.....1.239 DM	65 ■■■■ 28ms RLL.....1.799 ■■■■
43 MB 28ms MFM.....1.599 DM	

Alle Disks inkl. ALF 2.0 Software, Gehäuse, Netzteil  
Amiga Expansionsport ■■■■ durchgeführt, Anschlußkabel

### Preisgünstige Speichererweiterungen

abakus 512KB, ■■■■ Akku, abschaltbar.....	199 ■■■■
A ■■■■ 2MB-Card, Akku, abschaltb., 512KB bestückt.....	349 DM
A ■■■■ 2MB-Card, Akku, abschaltb., 512KB bestückt.....	DM
2-MB-Box, Akku, ab- u. umschaltb., 512KB bestückt.....	449 DM
4MB-Box, Akku, ■■■■ u. umschaltb., 2 ■■■■ bestückt.....	■ ■ ■ ■
MICROBOTICS 8-UP 8 MB bestückt.....	1.999 ■■■■

### Kickstart-Umschaltplatinen

Kick-Umschaltplatine ROM/ROM mit ■■■■ 1.2.....	69,90 DM
Kick-Umschaltplatine ROM/ROM mit ROM 1.3.....	99,90 DM
Kick-Umschaltplatine ROM-EPROM.....	69,90 DM
BootStrip, mit RAM, mit WOM, mit Gary Adap.....	339 ■■■■

Öffnungszeiten (Büro + Ladengeschäft)  
Mo - Fr 10.00 - 18.30 - Sa 10.00 - 14.00 - länger Sa 10.00 - 16.00  
Sedanstraße 136 \* 5000 Wuppertal 2  
Tel. 0202/50501500 \* Martin Kramer

### DONAU-SOFT 24 h-Schnellversand

### Neutrale Disketten

3.5" 2DD (100 % errorfree)

	von Sentinel	von SONY/Collossus
bis 99 Stück	1,60 DM	2,00 DM
ab 100 Stück	1,40 DM	1,85 DM
ab 500 Stück	1,25 DM	1,70 DM

### Laufwerke mit allen Extras

3.5" intern.....	155,- DM
3.5" extern, abschaltbar, Busdurchführung.....	209,- DM
5.25" extern, ■■■■ 3.5" + 40/80-Trackumschaltung.....	269,- DM

Sim City.....	89,- DM	GFA-Basic.....	208,- DM
B.A.D.....	77,- DM	GFA-Compiler.....	129,- DM
DPaint III ■■■■.....	240,- DM	512 KB-Erw. (A500).....	208,- DM
Zoetrope 1.1.....	189,- DM	2 MB-Erw. (A500).....	598,- DM

Vorkasse: + 5,- DM, Nachnahme: + 8,- DM, Ausland: + 10,- DM

### MAIK HAUER

Postfach 14 0 ■■■■ 858 Neuburg Fax: 0 84 31/4 98 00  
Tel.: 0 84 31/4 97 98 (bis 22 Uhr) BTX: \*Donau-Soft #

**D**ie Franz-Diskette Nr. 01 umfaßt sowohl kleine Spielchen als auch nützliche Tools. Außerdem bietet sie dem Programmierer verschiedene Programme, deren Sourcecodes ebenfalls auf der Diskette enthalten sind.

Das erste Programm heißt **Dazzle**, das sich auf einer der ersten Fish-Disketten befand. Es erzeugt mit Hilfe einer mirror (Spiegel) -ähnlichen Funktion sehenswerte Zufallsgrafiken, die sich ständig verändern. Der zugehörige C-Source ist auf der Diskette enthalten.

Bei **GFXMEM** handelt es sich um eine nützliche Speicheranzeige, die in C geschrieben wurde. Es wird in einem separaten Fenster der freie Speicher in Form von zwei Balkengrafiken dargestellt, die das Chip- und das Fast-RAM symbolisieren. Auch bei diesem Programm wurde der Source nicht vergessen.

#### Liebe Leser,

Die Programme, auf die wir im Rahmen der PD-Seiten eingehen, befinden sich größtenteils auch auf anderen PD-Reihen. Die Preise für die Disketten variieren jedoch von Vertreiber zu Vertreiber. Wenn Sie sich für ein bestimmtes Programm interessieren, sollten Sie in eigenem Interesse einen Preisvergleich anstellen.

**Clock** beinhaltet ebenfalls eine Anzeige über den freien Speicher. Außerdem stellt es zusammen mit der Speicheranzeige auch die aktuelle Uhrzeit in einem Fenster auf dem Titelfeld dar. Das nächste Programm auf dieser Franz-Diskette ist **KickChange**. Es bietet den Besitzern eines Amiga 1000 die Möglichkeit, die Kickstartversion zu wechseln, ohne jedesmal den Computer ausschalten zu müssen. Wer die Funktion dieses Programmes näher untersuchen möchte, findet den BASIC-Sourcecode dazu ebenfalls auf der Diskette.

Für C-Programmierer ist **Palette** interessant. Auch hier liegt der Sourcecode des in C geschriebenen Programmes auf der Diskette vor. Es dient dazu, die mit dem Programm Preferences eingestellten Farb-

## Die PD-Werkzeugkiste

**Nachdem wir uns in der letzten Ausgabe etwas mehr den Gag-Programmen gewidmet haben, wollen wir heute wieder einmal ein paar kleine Helferlein vorstellen, die den Umgang mit dem Amiga leichter gestalten sollen.**

werte der Workbench nachträglich zu ändern. Die Farben lassen sich durch einen Schieberegler einstellen und werden als Hexadezimalzahlen dargestellt.

**FontEdit** ist, wie der Name schon vermuten läßt, ein Zeichensatz-Editor. Er erlaubt es, beliebige Fonts (Zeichensätze) von Diskette einzuladen und nachzubearbeiten oder ganz einfach neue zu erstellen. Sowohl die Erstellung der Zeichen als auch die Bedienung erfolgt komplett mit der Maus. Das Programm bietet alle notwendigen Optionen, beispielsweise das Verändern der Baseline oder Breite und Höhe der Zeichen. Außerdem ist es möglich, proportionale Zeichensätze zu erstellen.

Nun aber zu einem negativen Punkt des Programms. Es kam beim Austesten des Zeichensatz-Editors mit einem AMIGA 2000 und KICKSTART 1.3 zu gelegentlichen Abstürzen. So findet man auch in der Anleitung den Vermerk, daß das Programm nur mit einem AMIGA 1000 und KICKSTART 1.1 ausprobiert wurde und nur mit dieser Anlage ein reibungsloser Betrieb garantiert werden könnte. Nichtsdestotrotz kann man mit dem Programm

aber über weite Strecken gut arbeiten.

Das Tool **MenuEditor** verspricht allen Besitzern der Betriebssystemversion KICKSTART 1.1 das einfache Erstellen eigener Menüs und das Einbinden in selbst erstellte C-Programme.

**MyCLI** stellt eine Ergänzung zum Original-CLI dar. Zum Beispiel erlaubt es die Belegung der Funktionstasten mit beliebigen Texten. Um die Taste F1 mit dem Text "dir df1:c/" zu belegen, schreibt man einfach den Befehl: DEF F1 dir df1:c/. Außerdem stellt MyCLI viele Befehle zur Verfügung. Die wichtigsten Kommandos stehen mit DEF, HELP, OFFLINE und TERMINAL ins Haus - pardon, in den Amiga. Mit dem letzten Befehl gelangt der Benutzer in einen Terminal-Modus, der es gestattet, direkt mit einem VAX-Rechner in Verbindung zu treten. Alle Tasteneingaben werden sowohl auf dem Bildschirm, als auch an der seriellen Schnittstelle ausgegeben. Der Befehl ENDCLI funktioniert mit KICKSTART 1.2 (oder höher) nicht immer einwandfrei. Ansonsten stellt MyCLI eine nützliche Hilfe dar.

Als nächstes Programm findet der PD-Interessierte **SuperBit-**

**Map**. Dabei handelt es sich um eine Demonstration der grafischen Fähigkeiten des Amiga. Es werden Linien auf eine übergroße Bitmap gezeichnet, und dabei ist es möglich, mit den Tasten u,d,l,r die Grafik in alle Richtungen scrollen zu lassen. Das Programm bietet etwas fürs Auge und für Programmierer. Wer der Hochsprache C mächtig ist oder, besser noch, nicht ganz mächtig ist, der findet auf der Franz-Diskette Nr. 01 das C-Sourcelisting dieses Programms.

Das letzte Programm auf der Franz-Diskette nennt sich **SpeechToy**. Bei diesem Programm handelt es sich um ein Sprachausgabeprogramm, bei dem sich alle Einstellungsmöglichkeiten über Gadgets und Schieberegler kontrollieren lassen. Als Besonderheit stellt SpeechToy die Möglichkeit zur Verfügung, ein Gesicht anzuzeigen, das beim Sprechen den Mund bewegt. Das Programm kann komplett mit der Maus gesteuert werden.

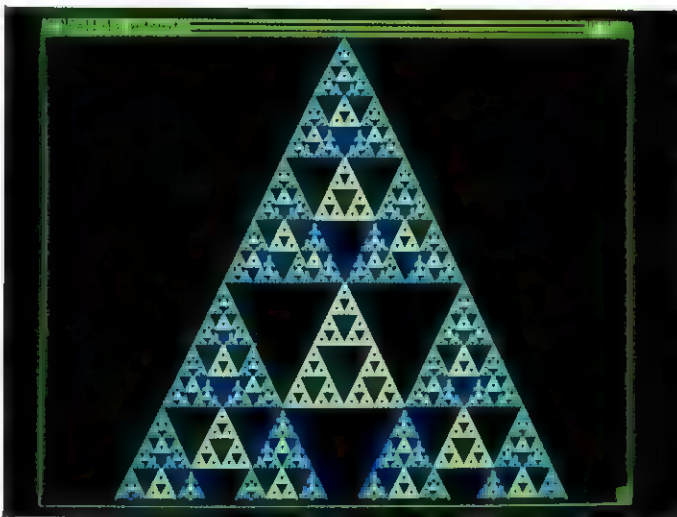
(Ingmar Reyer/br)

**Name:** Franz Nr.01  
**Quelle:** APS-Electronic

**D**ie Franz-Diskette Nr.47 ist besonders für Programmierer geeignet, die sich etwas in Assembler auskennen. Aber auch die Computerbenutzer, die sich für Mathematik interessieren, kommen nicht zu kurz.

### Beispielprogramme für den A68k-Assembler

Das Programm **Intro\_Maker** bietet Spieleprogrammierern die Möglichkeit, einen Vorspann mit soft-scrollendem Sternenhintergrund und Text zu erstellen. Der Text, der von rechts nach links bewegt wird, kann bis zu 5000 Zeichen enthalten. Mit diesem Tool wird dem Benutzer viel Zeit erspart, die man nunmehr für das Hauptprogramm zur Verfügung hat. Die Scrollmessage kann dann unter einem beliebigen Namen abgespeichert werden und ist dann eigenständig lauffähig. Im Verzeichnis A68K\_Bei-spiele findet der Assembler-



Das Programm Cell-Auto auf der Diskette Franz Nr.47 - so schön kann Geometrie sein



Fan einige Programme mit Quellcode. Das erste File heißt **Cell-Auto**. Es zeichnet sogenannte Sierpinsky-Tetraeder, also Grafiken, die sich aus Dreiecken aufbauen. Die Außenform ist ein großes Dreieck, das mit vielen kleineren gefüllt ist. Das Programm ermöglicht es, die Grafik mit fünf Parametern zu verändern.

**Lissajous** zeichnet gleichnamige Figuren. Bei diesem Programm ist besonders der Quellcode interessant, da es den Umgang mit der Mathtrans.library recht gut veranschaulicht.

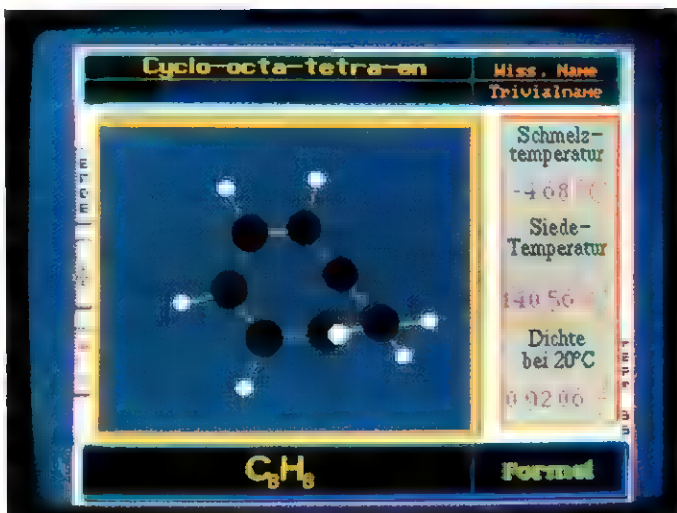
Weiterhin sehr interessant ist das Programm **Say**. Es liest dem gestreiften Benutzer ASCII-Dateien vor. Dabei wird immer eine Zeile separat auf dem Bildschirm angezeigt und vorgelesen.

Für Mathematiker ist wahrscheinlich **Prime** von Nutzen. Es gibt in rasender Geschwindigkeit Primzahlen in einem Fenster aus.

**Plane** hilft dem Programmierer beim Verständnis der graphics.library. Der Quellcode demonstriert das Linienzeichnen in Perfektion. Startet man das Programm, so wird ein Flugzeug gezeichnet, wieder gelöscht, dann um einen bestimmten Winkel gedreht und wiederum gezeichnet. Dieser Vorgang wiederholt sich ständig, bis das Programm abgebrochen wird.

Das Programm **Dump** ist ein äußerst nützliches Tool: es handelt sich hierbei um einen Speichermonitor, der den RAM-Speicher des Amiga als ASCII-Zeichen, als Hex-Zahlen oder in disassemblierter Form darstellt. So ist es möglich, einen tieferen Blick in das Betriebssystem oder auch in diverse Programme zu werfen. Die Bedienung stellt kein Problem dar. Die gesamte Eingabe läßt sich benutzerfreundlich mit der Maus steuern. Weiterhin bietet Dump viele Optionen. So kann der User nach bestimmten Zeichenfolgen suchen oder mit dem Menüpunkt "Micro" den verwendeten Mikroprozessor auswählen (68000, 68010 bzw. 68020).

**Boing** ist wieder etwas für Spieleprogrammierer. Der Quellcode wird dieser Gruppe helfen, einige Geheimnisse der Spriteprogrammierung zu lüften. Boing stellt zwei Sprites dar, die sich schnell über den Workbench-Screen bewegen.



Molekülstrukturen werden auf der Franz Nr.397 dreidimensional dargestellt. Die Moleküle können dabei auch animiert werden

Das letzte Demoprogramm für den A68K-Assembler nennt sich **Cmp**. Cmp ist aber nicht nur ein einfaches Demo. Nein, es ermöglicht das Vergleichen zweier Programme (ähnlich dem diff-Programm) und zeigt deren Unterschiede an. Bei der täglichen Arbeit am Computer ist dies natürlich sehr hilfreich. Hat man beispielsweise den Verdacht, einen Linkvirus auf seiner Diskette als Gast zu haben, benutzt man einfach Cmp und vergleicht alle Dateien mit denen auf einer (hoffentlich) angelegten Sicherheitscopy. Wird ein Virus geortet, kann man wirksam gegen ihn vorgehen.

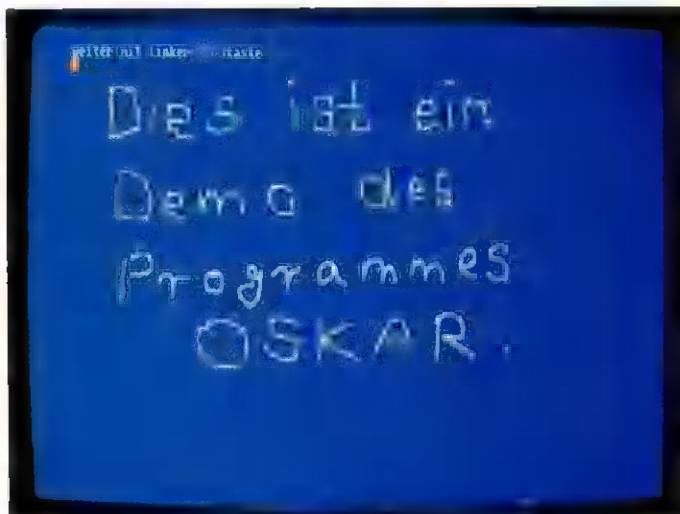
**Corral** ist ein Abkömmling der vielen Fraktal-Generatoren oder Apfelmännchenprogramme. Mit ähnlichen Formeln zeichnet er aber keine "dickbäuchigen Männer", son-

dern bizarre Korallenstrukturen. Die erstellten Grafiken sind wirklich sehenswert, aber "dank" der komplexen Formeln auch nicht gerade sehr schnell berechnet.

(Ingmar Reyer/br)

Name: Franz Nr.47  
Quelle: APS-Electronic

**E** fulminante Lücke im Softwarebereich besteht immer noch bei den Chemieprogrammen. Zwar wird mit solchen Programmen nur eine kleine Interessengruppe angesprochen, aber auch Außenseiter sollten auf ihre Kosten kommen. Dies sagte sich bestimmt auch Holger Franz, als er die Diskette Nummer 39 seiner Franz-Serie zusammenstellte.



Mit dem Programm Oskar von der Bavariansoft Nr.91 lassen sich nette Vorspanne erzeugen

So enthält diese Diskette ausschließlich ein Chemieprogramm namens **Organische Moleküle**. Nach dem Booten wird dem Benutzer die Möglichkeit geboten, eine RAM-Erweiterung zu nutzen. Mit dieser Erweiterung wird das ständige Nachladen der Moleküle vermieden. Vorab öffnet sich ein Titelbild mit der Aufschrift 'Organische Moleküle in Rotation'. Dem Titel nach zu urteilen, beschäftigt sich das Programm also mit der Darstellung dreidimensionaler Moleküle im Raum. Über den rechten Bildschirmrand erstreckt sich senkrecht ein Zahlenmenü, das von eins bis neun reicht. Hinter diesen Zahlen verbergen sich die einzelnen Moleküle, die in dem Programm vorkommen.

## Molekülstrukturen genauer betrachtet

Nach dem Anklicken einer Zahl wird das betreffende Molekül eingeladen und auf dem Bildschirm dargestellt. Dem Anwender bietet sich nun eine Benutzeroberfläche, in der er Näheres über das ausgewählte Molekül erfahren kann.

Inmitten des Bildschirms wird nach dem Anklicken eines Start-Requesters das betreffende Molekül räumlich animiert. Wie in der Chemie üblich, werden bei der Darstellung die Atome dabei mit verschiedenen Farben und Größen gekennzeichnet:

Kohlenstoff = schwarz  
Wasserstoff = weiß  
Sauerstoff = rot

Die Bindungen zwischen den einzelnen Atomen sind mit Strichen gekennzeichnet, so daß auch Doppel- bzw. Dreifachbindungen bei dieser Darstellung sofort erkannt werden können.

Die Geschwindigkeit, mit der sich das Molekül dreht, kann durch zwei Buttons verändert werden. Am unteren Rand des Bildschirms zeigt sich dem Benutzer die Strukturformel des Moleküls. Die dazugehörige Dichte, Schmelz- und Siedetemperatur findet man an der rechten Seite. Mit der Nennung des Trivial- und wissenschaftlichen Namens ist das Programm schließlich voll ausgekostet. Schade ist jedoch, daß man keinen Einfluß auf die Moleküle hat oder sich gar selbst welche erstellen kann.

Es sind lediglich neun Moleküle vorgegeben. Hier eine Liste der im Programm enthaltenen Moleküle:

- Cyclooctatetraen ( $C_8H_8$ )
- Isopropanol ( $CH_3-CHOH-CH_3$ )
- tert. Butanol ( $CH_3-C(CH_3)_2OH-CH_3$ )
- Cyclohexan ( $C_6H_{12}$ )
- Vinyllethin ( $C_4H_4$ )
- Dichlorbenzol ( $C_6H_4Cl_2$ )
- Acetonphenon ( $CH_3-CO-C_6H_5$ )
- Styrol ( $C_6H_5-CH=CH_2$ )
- Trimethylamin ( $(CH_3)_3N$ )

(Jürgen Seibel/br)

**Name:** Franz Nr.397  
**Quelle:** APS Electronic

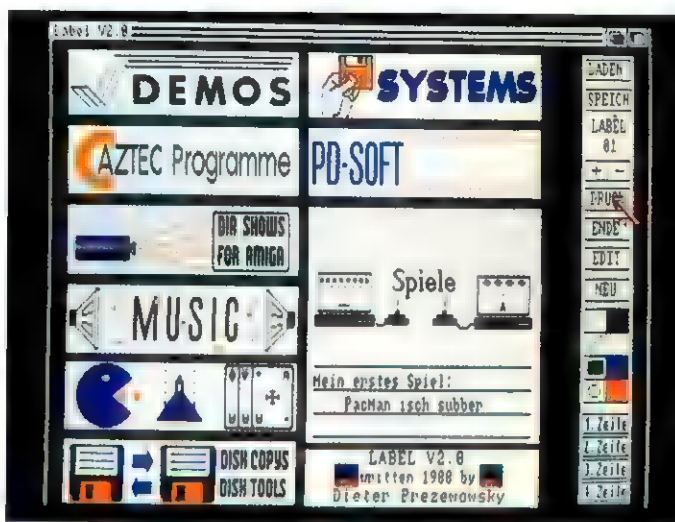
**E**s ist immer wieder überraschend, was den Amigabenutzern auf PD-Disketten angeboten wird. Das Thema der Bavariansoft Nr.40 ist das Ausdrucken von Etiketten. Dazu werden dem Benutzer sieben Etikettenprogramme zur Auswahl gestellt.

## Labels der Luxusklasse

Das Programm **Label V2.0** bietet ein hohes Maß an Funktionen. So stehen dem Benutzer anfangs acht verschiedene Diskettenlabel-Grafiken zur Verfügung. Auf der Diskette befinden sich jedoch weitere fünfzehn Bilder, die durch eine Einlade-Option angesprochen werden können.

Am rechten Bildschirmrand befinden sich die Optionen zum Erstellen eines Etiketts. Dabei spielt die Edit-Funktion die entscheidende Rolle. Nach dem Anklicken dieses Menüs können Sie eigene Label-Grafiken erstellen. Hierzu steht Ihnen, wie bei DPaint ein Malmodus zur Verfügung. Per Maus werden die Koordinaten für ein Rechteck, einen Kreis oder lediglich eine Linie definiert. Die Texteingabe ist unabhängig vom Edit-Mode. Nur wenn dieser Modus deaktiviert ist, kann mit der Eingabe begonnen werden. Als Drucker-treiber dienen die Preferences, so daß eine Druckeranpassung keine Probleme aufwerfen sollte.

Das nächste Etikettenprogramm namens **Etiketten V1.02** birgt nichts Aufregendes. Das Programm wird über drei auf dem Bildschirm erscheinende Schalter gesteuert.



Mit dem Etikettendruck-Tool Label V2.0 von der Bavariansoft Nr.40 lassen sich tolle Diskettenaufkleber erstellen

Diese Schalter dienen zur Eingabe, zum Drucken und zum Beenden des Programms. Ein Manko dieses Programms ist jedoch, daß die Daten nicht gespeichert werden können. Der **Disk-Label-Drucker** ist komplett in AmigaBASIC geschrieben. Je nach Speicherplatz gliedert sich das Programm. Im Falle einer Speichererweiterung können zusätzliche Optionen genutzt werden. Hierbei handelt es sich um zwei Programmfiles, die zum Hauptprogramm geladen werden. Sie ermöglichen das Umgestalten von Labelgrafiken. Dazu wird ein IFF-Bild mit einem Malprogramm (beispielsweise DPaint) bearbeitet. Nach dem Einladen des Hauptprogramms prüft der Rechner, welche Laufwerke angeschlossen sind. Danach erscheint ein Titelbild, das die Benutzeroberfläche darstellt. Durch das Anklicken diverser Menüs er-

reicht man die Texteingabe. Mit dem Anklicken des Bildmenüs öffnet sich ein Fenster, in dem sich die zur Auswahl stehenden Grafiken befinden. Die momentan ausgewählte Grafik ist umrahmt.

Das Programm **Etti** (nicht zu verwechseln mit unserem Listing aus Ausgabe 3/90) ist ebenfalls in AmigaBASIC geschrieben, jedoch enthält dieses Programm weit mehr Optionen als das eben beschriebene Etiketten V1.02. Durch Tastaturkommandos können die folgenden Optionen ausgeführt werden:

Drucken, Color, Speichern, Laden, Verzeichnis, Reset, Quit, Text eingeben, Farbe ändern, Art der Schrift, Unterstreichen, Up and Down.

Das Verzeichnis Option dient zum Erstellen eines neuen Directories, dessen Name frei wählbar ist. Weitaus interessanter ist jedoch der Color-Be-

fehl. Aufgrund dieses Aufrufes können Sie endlich die Kapazitäten eines (eventuell vorhandenen) Farbdruckers nutzen. So kann man die Zeilen in den Farben schwarz, rot, blau, violett, gelb, lila und grün erscheinen und ausdrucken lassen. Ein wenig Pep wird nicht allein mit Farben eingeführt, sondern auch durch das Wählen der verschiedenen Schriftarten.

Mit **Etikettendruck** reiht sich ein weiteres Programm in diese Gruppe ein. Das Programm wird über ein Pull-down-Menü bedient, das über vier Hauptmenüs verfügt (Optionen, Druckfunktionen, Run und Windows). Im ersten Menü werden das Directory und die Unterverzeichnisse angezeigt. Zusätzlich kann das Programm in diesem Menü beendet werden. Das Druckfunktionen-Menü enthält einige nicht zu verachtende Optionen. Beispielsweise läßt sich hier der Drucker justieren. Dazu wird der Drucker in Bereitschaft gesetzt, worauf dann ein Probendruck zum Drucker geschickt wird. Des weiteren übergibt man in diesem Menü die Parameter zum Ausdruck. Hier folgt nach der Eingabe des linken Randes die Abfrage der Zeilenanzahl. Nach der Definition der Schriftart ist die Eingabe beendet. Auch bei diesem Programm besteht die Möglichkeit, die Schriftarten für jede Zeile zu variieren. Der Window-Befehl läßt das gesamte Listing des Programms auf dem Bildschirm erscheinen, so daß einzelne Routinen des Source-Codes nach Wunsch umgeschrieben werden können.

Von **Labelprint** existieren zwei Versionen: Labelprint und LabelprintE, die jedoch in ihrer Funktionsweise völlig identisch sind. Ihr einziger Unterschied liegt bei den Druckertreibern. LabelprintE ist für alle Epson-kompatiblen Drucker geschrieben. Dagegen spricht Labelprint den Drucker mit dem PCprtCommand-Befehl an. Bei LabelPrint schießt einem geradezu die Anzahl der Laufwerke, die unterstützt werden können, in die Augen. Man kann in diesem Programm auf vier Laufwerke, die RAM-Disk und auf eine Harddisk zurückgreifen. Ein residenter Search-Befehl sortiert die Fileliste vor dem Ausdruck ständig. Mit **Eti** präsentiert sich das letzte auf der Diskette befindliche Programm, das sich nur aus dem CLI starten



Auch das Programm BootIntro ist ein nettes Utility, mit dem man Scrolltexte automatisch beim Booten ausgeben kann



läßt. Eti ist ein Etikettendruckprogramm, das zwar spartanisch aufgebaut, aber dennoch alle wichtigen Fähigkeiten eines Label-Druckprogrammes beinhaltet. So öffnet sich nur ein Fenster, in das der Text eingegeben wird. Im Projekt Menü kann man die Labels dann abspeichern und in Schönschrift ausdrucken lassen.

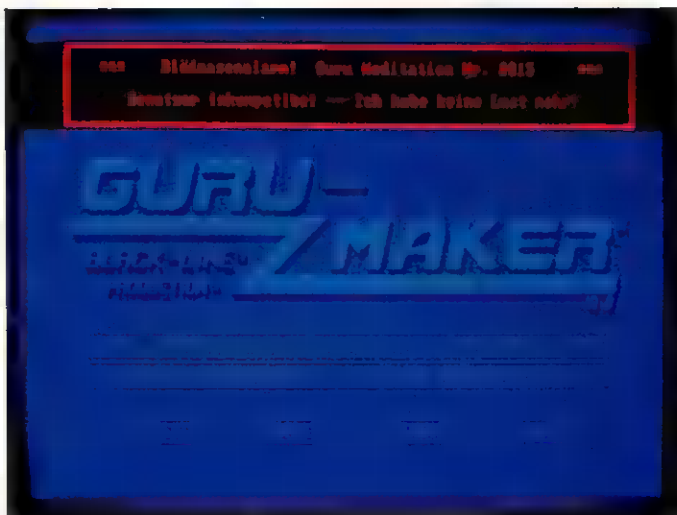
(Jürgen Seibel/br)

**Name:** Bavariansoft Nr.40  
**Quelle:** Bavariansoft

**B**ootblöcke werden gerne dazu genutzt, Messages beim "Hochfahren" auf den Bildschirm zu bringen. Dies hat einen großen Vorteil gegenüber normalbootenden Disketten, da im Falle eines Bootblock-Virusbefalles die erwartete Message nicht mehr erscheint. Auf der Bavariansoft Nr. 91 sind nun einige Utilities, die das Erstellen von Bootblöcken und Programmvorspannen zum Kinderspiel werden lassen.

## Bootblöcke und Programmvorspanne im Eigenbau

Sicher kennen Sie Oskar, den schnellsten Zeichner aus dem Fernsehen. Aber nicht nur der Fernsehzeichner kann schnell zeichnen – nein, auch der Amiga kann's. Das Programm **Oskar** (wie der Programmierer nur auf diesen Namen gekommen ist...) ist ein kleines Vorspannprogramm, das eine mit der Maus erstellte Grafik (oder



**Guru-Meditations simulieren – mit dem Guru-Maker ist dies kein Problem**

Text) auf dem Bildschirm ausgibt. Das Programm wird aus dem CLI mit dem Befehl **OSKAR Dateiname** aufgerufen, worauf man mit der Maus einen beliebigen Text (oder auch eine Grafik) schreiben bzw. malen kann. Das Programm erzeugt nun eine Datei, die mit dem CLI-Befehl **TRICK Dateiname DELAY** das Gezeichnete wiedergibt. Mit dem Zusatz **DELAY** hinter dem Dateinamen wird das Maß der Verzögerung angegeben, mit der die Datei auf den Bildschirm gebracht wird.

Bei dem Programm **Marquee** handelt es sich um einen Titelgenerator für Laufschriften. Man erzeugt mit einem beliebigen Texteditor (beispielsweise dem ed des AmigaDOS) eine ASCII-Datei und speichert sie unter einem beliebigen Namen ab. Nach Aufruf der Datei **MakeText** muß zunächst der komplette Pfad zum Marquee-File und dann die Textdatei

angegeben werden. Ist dies geschehen, so öffnet sich nach kurzer Zeit ein Bildschirm, in dem der Text als Laufschrift gescrollt wird, dieser wird zudem mit einem Musikstück (der Entertainer aus dem Film 'Der Clou') untermalt.

Mit dem **Bootwriter** wird ein Programm im Bootblock einer Diskette abgelegt, bei dem sowohl der Hintergrund als auch die Laufschrift gescrollt wird. Die Laufschrift kann dabei frei editiert werden. Der Programmaufruf erfolgt aus dem CLI, worauf der Benutzer aufgefordert wird, den Hintergrund festzulegen. Hierbei stehen dem Anwender verschiedene Hintergrundgrafiken zur Verfügung. Nun braucht nur noch der Lauftext eingegeben und abgespeichert zu werden und fertig ist der neue Bootblock.

**BootIntro** ist ein Bootblock-Utility, das wir schon in der Ausgabe 2/90 vorgestellt hat-

ten. Auf einem farbigen Hintergrund wird eine feststehende Kopfzeile und ein horizontal scrollender Fließtext dargestellt. Die Überschrift kann maximal 20 Zeichen und die Laufschrift maximal 225 Zeichen umfassen.

Der nächste Display-Hack nennt sich **Startle** und basiert auf dem Programm Starts von Leo Schwab. Zuerst erstellt man sich bei diesem Programm eine beliebige Textdatei und speichert sie ab. Der Aufruf von Startle erfolgt wiederum aus dem CLI mit der Syntax **startle <Textdatei>**, wobei Parameter für den Zeichensatz und die Zeichenfarbe ergänzt werden können. Der Text scrollt von unten nach oben über den Bildschirm, während der Benutzer den Anschein hat, er würde durch den Weltraum fliegen.

**Guru-Maker** ist das letzte Programm auf der Bavariansoft Nr.91. Mit diesem Tool können Sie die Ausnahmezustände (besser bekannt als Guru-Meditations) simulieren. Zwei Eingabezeilen stehen dabei für den Text zur Verfügung, der maximal 27 Zeichen lang sein darf. Durch Aktivieren eines Test-Buttons kann sich der Benutzer sein Werk vorab ansehen. Laut Autor kann man sich diese Message auf dem Bootblock abspeichern, die dann beim Booten der Diskette auf dem Bildschirm erscheint. Diese Funktion versagte jedoch bei unserer Version den Dienst.

(br)

**Name:** Bavariansoft Nr.91  
**Quelle:** Bavariansoft

## Eine kleine Übersicht der Vertreiber von Public Domain, Free- und Shareware sowie Prüf-vor-Kauf-Programmen (ohne Anspruch auf Vollständigkeit)

A.P.S. Electronic  
Sonnenborstel 31  
3071 Steimbke  
Tel.: 05026/1700

BAVARIANSOFT  
Postfach 72  
8473 Pfreimd  
Tel.: 09606/286

BELLINGRATH,  
CHRISTIAN  
Hans-Böckler-Str. 55  
5860 Iserlohn  
Tel.: 02371/24192

DIGITAL MARKETING D.  
MÜCKTER  
Krefelder Str. 16  
5142 Hückelhoven-Baal  
Tel.: 02435/2086,428  
oder 1295

HIESKE, DIETER  
Schillerstr. 36  
6700 Ludwigshafen  
Tel.: 0621/673105

KEIM, PETER  
Vogelsanger Str. 34  
5000 Köln 30  
Tel.: 0221/520765

MAXON COMPUTER GMBH  
Industriestr. 26  
6236 Eschborn  
Tel.: 06196/481811

Amiga PD-Depot  
Wolfgang Bittner  
Keltenstr. 15  
6700 Ludwigshafen  
Tel.: 0621/674974

TechnikSupport Verlag  
GmbH  
Bundesallee 36-37  
1000 Berlin 31  
Tel.: 030/8621314

WOLF Computertechnik  
Deipe Stegge 187  
4420 Coesfeld  
Tel.: 02541/2874

**Österreich**  
KÜPPERS, BERND  
Felberstr. 7  
A-5730 Mittersill  
Tel.: 06562/282

PETER RAUSCHERS  
COMPUTERSHOP  
A-1100 Wien  
Weldengasse 41  
Tel.: 0222/621535

In den USA ist das Spiel Core Wars auf PCs und Macintoshs schon weit verbreitet, in Europa ist es jedoch noch so gut wie unbekannt. Dies soll mit der Amiga-Version als Public-Domain-Software anders werden.

Bei dem Spiel handelt es sich um ein Programm, das "Kampfprogramme" gegeneinander antreten läßt und diesen Vorgang grafisch darstellt. Der Begriff "Kampfprogramm" bedarf wahrscheinlich einer Erklärung: Diese Programme werden in einer verkürzten Assemblersprache geschrieben.

## Core Wars

Bei Core Wars nennt sich die Assemblersprache Redcode und umfaßt zehn Befehle. Mit Hilfe dieser Befehle ist es möglich, Speicheradressen zu bombardieren, das heißt, bestimmte Adressen werden mit Nullen überschrieben. Außerdem kann man sein eigenes Programm im Speicher verschieben oder reparieren. Wenn man mit diesen Möglichkeiten geschickt umgeht, kommen dabei erstaunliche Programme heraus. Sie haben zum Ziel, ein anderes, ebenfalls im Speicher stehendes Programm zu zerstören. Sie werden sich jetzt fragen: Was haben denn dabei die Mitspieler zu tun? Die Antwort ist einfach. Der Computer spielt gegen sich selbst das von den Programmierern (Spielern) erstellte Programm. Er arbeitet dabei zwei Kampfprogramme abwechselnd und schrittweise ab. Das interessante daran ist, zu sehen, wie das eigene Programm beim Kampf abschneidet. Es ist schon toll, wenn man sieht, daß das in stundenlanger Arbeit erstellte Programm das des Gegners im Kampf besiegt.

Im Programmieren Ungeübte brauchen jetzt aber nicht den Kopf hängen zu lassen. Es stellt selbst Anfänger vor keine großen Probleme, mit Hilfe der sehr ausführlichen und guten Beschreibung (80 Seiten) kleinere Programme schon nach einer halben Stunde selbst zu erstellen. Core Wars enthält einen schnellen Editor, mit dem man

**Name:** Core Wars  
**Quelle:** Panorama 30b,  
Franz 21

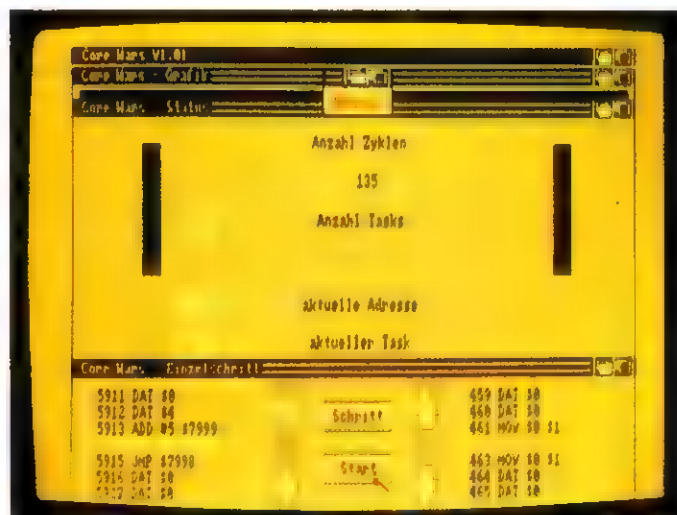
# Public-Domain-Spieleshow

**Nicht immer haben Public-Domain-Spiele kommerzielle Vorbilder. Andere Produkte orientieren sich an Brettspielen – Themen, die für die Softwareindustrie erst bei Bestsellern wie Trivial Pursuit interessant werden. Public-Domain-Programme entstehen zumeist aus Spaß an der Freude. So gibt es für Public-Domain-Anwender viele kleine Leckerbissen, mit denen der kommerzielle Softwaremarkt nicht aufwarten kann.**

sehr komfortabel programmieren kann, einen Assembler, der die Befehlsfolgen in für den Computer verständliche Befehle übersetzt, und ein Ablaufprogramm, mit dem es möglich ist, zwei Kampfprogramme gegeneinander antreten zu lassen. Das Ablaufprogramm stellt den Speicher (das Spielfeld) grafisch dar.

Damit ist leicht nachzuvollziehen, wie die Programme arbeiten.

Ein, nach wie vor sehr beliebtes Brettspiel ist Risiko. Nun gibt es eine Public-Domain-Version für den Amiga, die sehr gut gelungen ist. Das Spielfeld



Krieg der Programme – bei Core Wars wird der Speicher Ihres Amigas zum Schlachtfeld



Eine Amiga-Umsetzung des beliebten Risiko-Brettspiels

von Risk besteht aus einer Weltkarte mit sechs Kontinenten und 42 Ländern, die auf bis zu fünf Mitspieler aufgeteilt werden.

## Risk

Von diesen fünf Mitspielern kann der Computer beliebig viele simulieren. Pro Spielrunde erhält jeder Spieler zusätzliche Truppen, um möglichst viele Länder besetzt zu halten oder erobern zu können. Je mehr Länder von einem Spieler besetzt sind, desto größer ist die Anzahl neuer Truppen, die er pro Zug hinzubekommt. Dies ist auch bitter nötig, da die Truppenstärke bei jedem Durchgang stark dezimiert wird. Das Spiel hat der Gewinner, der es schafft, alle Länder zu besetzen oder eine zu Beginn des Spieles gestellte Aufgabe zu lösen. Bei den Kämpfen, die dabei unerlässlich sind, spielt der Zufall in Form von Würfeln mit. Je mehr Truppen ein Land angreifen oder es besetzen, desto mehr Würfel hat der Spieler. Allerdings sind drei Würfel das Maximum. Der, der die größte Augenzahl erreicht, bezwingt eine Truppeneinheit des Gegners. Das Würfeln wird so oft wiederholt, bis der Angreifer oder der Besitzer des Landes keine Truppen mehr im umkämpften Gebiet hat.

Risk sorgt mit seinem taktischen Element und der Fülle einstellbarer Optionen für lange anhaltenden Spielspaß. Die Mausbedienung des Programms ist problemlos, und die Grafik ist schön gezeichnet. Ein Minuspunkt muß dennoch angemerkt werden. Es ist mit 512 kByte Speicher nicht möglich, das Programm von der Workbench zu starten.

**Name:** RISK  
**Quelle:** Oase

Würden Sie gern das Handeln mit Aktien trainieren? Sie glauben, Ihnen fehlt das nötige Geld dazu? Mit dem Simulationsprogramm Broker ist es möglich, ohne eigenes Kapital das richtige Spekulieren mit Aktien zu üben.

Broker bietet den bis zu vier Mitspielern die Möglichkeit, Aktien zu erstehen und abzu-





**Aktien und Wertpapiere – darum dreht sich an der Börse und in dem PD-Spiel Broker alles**

stoßen. Dabei müssen sie Steuern, Optionen, Dividenden und Kontozinsen berücksichtigen. Weiterhin kann jeder Spieler einen Kredit aufnehmen, den er plus Zinsen im Verlauf des Spieles abzahlen muß. Die Kurse der Aktien berechnen sich aus den Voraussagen, die die Spieler zuvor abgegeben haben.

## Broker

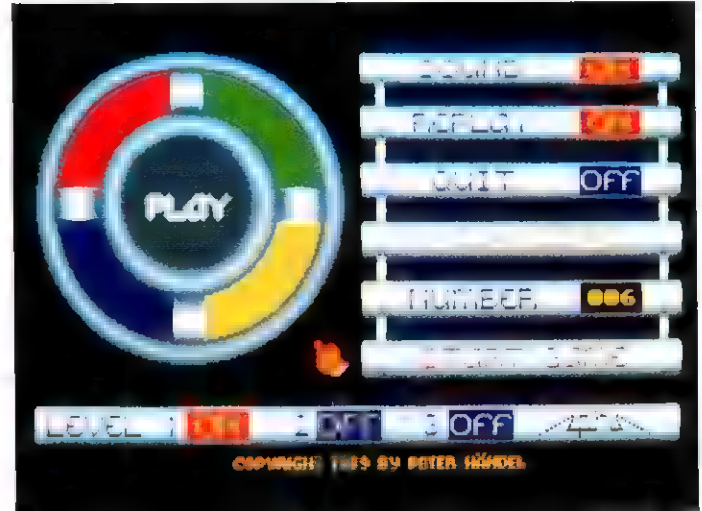
Soll eine Aktie im Wert steigen, wählt man den Menüpunkt: "Aktie steigt im Wert mit x-prozentiger Wahrscheinlichkeit". Das Programm stellt verschiedene Prozentwerte zur Verfügung. Je höher dieser Wert gewählt wird, desto mehr Punkte muß der Spieler dafür opfern. Zu Beginn stehen jedem Mitspieler 1000 dieser Punkte zur

Verfügung. Das klingt im ersten Moment viel, aber beim Versuch, seine eigenen Aktienkurse anzuheben, sind sie schnell aufgebraucht.

Optisch bietet das Programm naturgemäß wenig. Doch reizt es durch die interessante Thematik und die spannenden Finanzschlachten, die sich zwischen mehreren Mitspielern ergeben können.

**Name:** Broker  
**Quelle:** Oase

**O**hne ein gutes Gedächtnis werden Sie bei Senso wohl keine besonders guten Ergebnisse erzielen. Es geht darum, eine Folge aus verschiedenen Tö-



**Gedächtnistraining spielerisch – es gilt, Ton und Farbfolgen fehlerfrei zu reproduzieren**

nen nachzuspielen. Jedesmal, wenn eine Tonfolge vom Spieler komplett richtig reproduziert wurde, wird sie durch eine neue, längere Tonfolge ersetzt.

## Senso

Insgesamt gibt es vier verschiedene Töne, die sich in vier verschiedenfarbigen Feldern widerspiegeln. Jedesmal wenn ein Ton ertönt, leuchtet das entsprechende Farbfeld parallel dazu auf. Drehen die Spieler den Ton ab, können anstelle der Tonfolgen die Farbfolgen erraten und reproduziert werden.

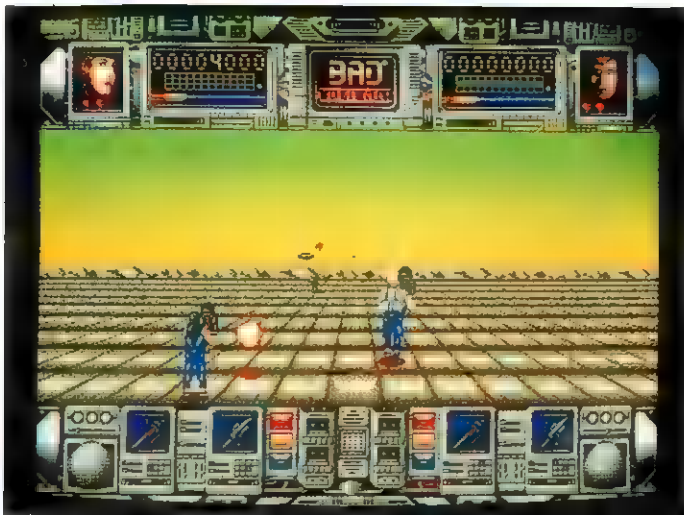
Senso ist ein einfaches, aber vernünftig gemachtes Gedächtnisspiel, das insbesondere in Gesellschaft viel Spaß macht. Der Autor Peter Händel zeichnete schon in der Vergangen-

heit für einige andere, wirklich gelungene Spiele verantwortlich (Steinschlag, Marble Slide etc.). Auch er scheint über die Art, wie in der Vergangenheit mit den Programmierleistungen enthusiastischer PD-Autoren Geschäfte gemacht wurden (und nach wie vor gemacht werden), unzufrieden gewesen zu sein. Er bezeichnet sein Programm in aller Deutlichkeit als Freeware und besteht im Vorspann auf eine nichtkommerzielle Weiterverbreitung. Mit Fug und Recht, wie wir meinen.

(Ingmar Reyer/hs)

**Name:** SENSO  
**Quelle:** Franz  
**Besonderheit:** Freeware





Als Kosmo-Söldner auf Alienhatz, keine weiteren Fragen...

## bad company

Wenn es einen Preis für Ehrlichkeit gäbe, so hätte ihn Bad Company redlich verdient! Was hat dieses Ballerspiel also zu bieten?

Zunächst einmal eine der aufrichtigsten Stories, die das Killen von Aliens rechtfertigt: Die Erde braucht neuen Lebensraum, die (ohnehin insektoiden) Aliens stehen dem im Wege. Also fort mit ihnen – natürlich zum Wohle der Menschheit, gelle? Für die Säuberung steht eine Truppe von acht Söldnern zur Verfügung, aus denen der Spieler seinen Weltraum-Rambo wählen darf. Insgesamt sind vier Planeten zu 'reinigen', auf denen neben Horden von Aliens auch Waffennachschub und Lebensenergie zu finden sind. Stehen genügend Joysticks zur Verfügung, können sich auch zwei Spieler auf die fröhliche Hatz begeben.

Tja, das war's dann auch schon. Das dreidimensionale Szenario scrollt in Fluchtpunktperspektive auf den Spieler zu, der braucht nichts weiter zu tun, als aufs Knöpfchen zu drücken und den gegnerischen Schüssen auszuweichen. Die Steuerung ist brauchbar, die Grafik guter Durchschnitt, nur vom Sound von David Whittaker wünscht man sich mehr (und vielleicht auch Besseres). Da jeder Kämpfer bestimmte Vorzüge hat und das Arsenal an Waffen reichhaltig ist, ist die Motivation anfangs recht hoch – allerdings nur am Anfang. Es

sei noch ein Zitat aus dem Covertext gestattet, der wie die Faust aufs (Alien-)Auge paßt: 'Fragen werden hier keine gestellt, und Antworten verhallen im Leeren. Es gibt nur eine Regel: Vernichtung der Aliens.' In diesem Sinne: Feuer frei... (Michael Anton/hs)

### AMIGA DOS Blitzlicht

**Name:** Bad Company  
**Hersteller:** Logotron Entertainment  
**Vertrieb:** Fachhandel  
**Preis:** 64,95 DM

**Positiv:**  
– ehrliche Story  
– Option für zwei Spieler

**Negativ:**  
– zu ehrliche Story  
– keine Neuheiten



Packen Sie Taktgefühl und Fairness besser wieder ein, hier wird mit dem Granatwerfer argumentiert

## 5th GEAR

Autorennen einmal anders – das bietet 5th Gear. Es fragt sich nur, ob diese Art von Rennen wirklich so gut ist...

Vergessen Sie alle Regeln der Fairness, Ihre mühsam absolvierten Fahrtstunden – und vielleicht auch die Ansprüche, die Sie an Amiga-Spiele stellen. Denn bei 5th Gear können Sie den Gegner mit MG oder Rakete von der Strecke pusten, mit absolut fehlgeplanten Strecken rechnen – und den Rest getrost vergessen...

Der Hintergrund wird nicht näher erläutert: Sie nehmen an einem illegalen Rennen teil, bei dem alles erlaubt ist, was verfüg- und finanzierbar ist. Diverse Shops auf der Strecke bieten neben dem autospezifischen Service wie Reparatur und Benzin alles, um aus dem Raser einen Kampfpanzer zu machen: Schnellfeuerwaffe und Raketenwerfer gibt es, mit Boostern kann man die Karre auch für kurze Zeit zum 'Flugzeug' machen. Mit dieser Ausrüstung geht es dann auf Strecke, eine in der Vogelperspektive dargestellte Wüstenlandschaft, in der es von Hindernissen und Gegnern nur so wimmelt.

(Offensichtlich wurde vergessen, das Auto mit einer effektiven Bremse auszurüsten.) Gewinnt man dennoch ein Rennen, gibt es Geld für neue Ausrüstung und eine neue Runde mit neuem Limit.

Die grafische Gestaltung ist akzeptabel, Steuerung und Sound während des Spiels könnten besser sein. 5th Gear ist ein Spiel, welches kurzzeitig Spaß macht, auf längere Sicht aber zu wenig bietet. Schade, denn man hätte mehr aus dieser Idee machen können... (Michael Anton/hs)

### AMIGA DOS Blitzlicht

**Name:** 5th Gear  
**Hersteller:** Hewson  
**Vertrieb:** Fachhandel  
**Preis:** 64,95 DM

**Positiv:**  
– fairer Preis

**Negativ:**  
– hakelige Steuerung  
– stellenweise unfair







Unterwasseraction hat Konjunktur, Aquanaut verfolgt in der Tiefsee Aliens

## Aquanaut

Da weder den Aliens noch den Programmierern Neues einfällt, ist die Erde mal wieder in Gefahr. Oder gibt's doch was Neues?

**E**her zufällig ist die für das 21. Jahrhundert geplante Invasion der bösen Ramanishi schief gegangen. Da so was keine gute Story abgibt, haben die Herren von Prism wenigstens einen Alien auf die Erde gelangen lassen, damit zumindest etwas zu machen ist... Besagter Alien hat sich inzwischen auf dem Meeresgrund niedergelassen und macht dort - ja, was macht er dort? Das weiß keiner so recht, es wird auch aus der Story nicht klar. Aber rein prophylaktisch sollte man ihm wohl mal auf die Finger klopfen, vielleicht erfährt man dabei ja etwas Neues. Seinen Kopf hierfür muß ein Marinetaucher herhalten, der ins eiskalte Wasser gestoßen wird. Dort findet er neben Kapseln mit Ausrüstung auch jede Menge mutierter Meeresfauna, die ihm ans Leder will. Durch drei Szenarien muß er sich quälen, bis die Menschheit gerettet ist: Das offene Meer, ein Labyrinth unter Wasser und eine geheimnisvolle Stadt warten auf ihre Erforschung...

Grafik und Sound könnten manchmal etwas besser sein, alles in allem besitzt das Spiel jedoch seine Reize. Sehr lobenswert ist auch, daß eine einmal gelöste Sequenz nicht immer neu gespielt werden muß. (Weniger gut ist,

daß darüber auf der Originaldiskette Buch geführt wird, Paßwörter wären zur Virenabschreckung doch besser gewesen.) Schade auch, daß der Rechner nach Spielende abgeschaltet werden muß, ein Reset ist nicht möglich...

(Michael Anton/hs)

AMIGA DOS Blitzlicht	
<b>Name:</b> Aquanaut	
<b>Hersteller:</b> Prism	
<b>Vertrieb:</b> Fachhandel	
<b>Preis:</b> 84,95 DM	
<b>Positiv:</b>	<ul style="list-style-type: none"> <li>- Spiel komplex und abwechslungsreich</li> <li>- Zwischenstand speicherbar</li> </ul>
<b>Negativ:</b>	<ul style="list-style-type: none"> <li>- Grafik und Sound etwas dürrig</li> <li>- Preis etwas hoch</li> </ul>



Auch bei diesem Fußballspiel kann es zu spannenden Zweikämpfen kommen

## Bodo Ilgners Super Soccer

So manchen packt bei der Übertragung eines heißen Bundesligaspiels die Lust, einmal selbst dem runden Leder nachzujagen. Und damit man sich nicht überanstrengt, gibt es die Fußballsimulationen.

**F**ußballspiele wurden schon oft und in allen Variationen auf Heimcomputern realisiert. Da gab es zum Beispiel Programme, in denen einfach nur gespielt wurde oder andere, in denen der Spieler die Rolle des Managers übernahm, und Mischungen aus beidem. Bodo Ilgners Super Soccer gehört zu letzteren. In den Menüs können Sie sich eigene Mannschaften, Ligen und Pokale entwerfen oder Ihr Team in verschiedenen Situationen trainieren. Außerdem hat jeder Spieler Eigenschaften wie Können, Kraft, Charakter und Spielstil und ein von Ihnen veränderbares Outfit wie Haarlänge und Farbe, Teint und Name.

Im eigentlichen Spiel können zwei Spieler gegeneinander oder ein Spieler gegen den Computer spielen. Beim Schießen geben sogenannte Boot-o-Meter Schußstärke und Höhe an.

Viel Verwirrung verursacht die Darstellung des Spielfeldes: Befindet sich der Ball im Mittelfeld, so wird der Platz in der Seitenansicht gezeigt. Kommt er allerdings in Tornähe, so wird das Tor von vorn gezeigt, so daß man nun von unten nach oben spielt. Kommt der Ball dann wieder

ins Mittelfeld, geht's wieder seitwärts weiter.

Bodo Ilgners Super Soccer gehört eher zu den schlechten Vertretern seines Genres, denn weder Grafik, Sound oder das Spielgeschehen können überzeugen.

(Robert Marz/hs)

AMIGA DOS Blitzlicht	
<b>Name:</b> Bodo Ilgners Super Soccer	
<b>Hersteller:</b> Empire Software	
<b>Quelle:</b> Fachhandel	
<b>Preis:</b> zirka 84,95 DM	
<b>Positiv:</b>	<ul style="list-style-type: none"> <li>- viele veränderbare Parameter</li> <li>- eigene Ligen und Pokale möglich</li> </ul>
<b>Negativ:</b>	<ul style="list-style-type: none"> <li>- wechselnde Perspektive</li> <li>- schwache Grafiken</li> </ul>

# AMIGA-DOS-Spieletips

**K**ennen Sie den lebenswerten und etwas tolpatschigen Helden aus dem Abenteuer "Future Wars - Time Travelers"? Der hat nämlich den ersten Teil seines schweren Abenteuers hinter sich und befindet sich zur Zeit irgendwo in der Zukunft. Bevor er sich an die Erkundung der für ihn unbekannten Gegend macht, beschließt er zunächst einmal, seine bisherigen Erlebnisse für die Nachwelt in einem Tagebuch festzuhalten. In der AMIGA DOS können Sie seine Eintragungen lesen.

## Zukunftskriege - die Abenteuer eines Fensterputzers

"...Die ganze Geschichte nahm beim Fensterputzen ihren Lauf. Ich weiß noch, wie ich, ausgerechnet als mein Boß aus dem Fenster schaute, den Putzeimer mit dem Fuß umstieß. Auf dieses Mißgeschick folgte eine lange Predigt seinerseits. Dabei regte er sich so sehr auf, daß er sein

**Viele heiße Tips für harte Spiele haben wir aus unserer großen Kiste mit den Spiellösungen herausgesucht und für diese Helpline zusammengestellt. Auch diesen Monat können wir dank der freundlichen Unterstützung durch Sie, liebe Leser, wieder viele Spielrätsel aufklären. Wenn Sie sich an der Helpline beteiligen wollen, auch Ihre Tips sind willkommen und werden bei der Veröffentlichung mit einem Software-Dankeschön honoriert.**

Fenster nicht richtig schloß. Ich nahm also meinen Eimer und schaute mir meine Plattform etwas genauer an. Auf dem Schalter drückte ich den Knopf für "rauf" und fuhr auf die Etage mit dem nicht fest verschlossenen Fenster. Nach dem Öffnen war der Weg ins Innere des Gebäudes frei. Eigentlich wollte ich nur meinen Eimer im Bad wieder füllen, aber einmal hier eingedrungen, beschloß ich, mich in der Chefetage etwas genauer umzusehen. Als erstes fiel mir der Papierkorb ins Auge. Er enthielt eine leere Plastiktüte, die ich in weiser Voraussicht an mich nahm. Im Bad wurde der Eimer am Waschbecken gefüllt. Im Schrank stand eine Flasche mit Insektizid, die in meinen Besitz

überging. Im WC fand ich dann noch ein kleines rotes Fähnchen. Nachdem ich alle Türen wieder verschlossen hatte, durchquerte ich den Raum, um ihn durch die Tür nach rechts zu verlassen. Auf meinem Weg stolperte ich über eine Beule im Teppich. Die Ursache dafür war ein kleiner Schlüssel, der darunter lag. Dann bekam ich ein Problem: Jedesmal, wenn ich die Tür auch nur berührte, öffnete sich die Tür meines Chefs, und er ließ ein erneutes Donnerwetter los. Ich tat dann lieber so, als wollte ich ganz schnell wieder verschwinden. Nach einiger Zeit kam mir eine geniale Idee: Ich positionierte meinen vollen Eimer direkt über seiner Tür. Von Vorfreude gepackt mach-

te ich mich auf den Weg zur rechten Tür. Gleich nach dem Öffnen, trat auch der Boß aus seinem Zimmer und der Eimer traf sein Ziel exakt. Schnell huschte ich ins nächste Zimmer. Ich schloß die Tür und nahm die Einrichtung in Augenschein. Einer der Schränke, der sich mit dem Schlüssel, den ich unter dem Teppich gefunden hatte, öffnen ließ, enthielt eine Schreibmaschine, mit deren Carbonfarbband eine Nummer geschrieben wurde, die ich mir glücklicherweise merkte. In der Schreibtischschublade habe ich dann noch einen Stapel Papier gefunden. Als nächstes nahm ich die Wandkarte in Augenschein, auf der mir besonders ein kleines Loch auffiel, in das das rote Fähnchen paßte. Darauf gab die Karte eine Geheimtür frei.

Neugierig betrat ich den fremd anmutenden Raum dahinter. Ich bemerkte mit Entsetzen, daß sich nach meinem Betreten die Decke langsam senkte. Die einzige Hoffnung bestand in der Tastatur an der Wand, mittels derer ich so schnell wie möglich die Nummer aus der Schreibmaschine eingab. Welch ein Glück, daß mein Einfall richtig war! Die Decke hob sich wieder und die Wand vor mir gab einen Durchgang frei, hinter dem sich ein Raum mit seltsamen Geräten befand. In der Mitte des Raumes stand eine Maschine, die wie ein Kopierer aussah. Ich drückte auf den grünen Knopf und die Maschine erwachte zum Leben, was sich in einem tiefen Brummen äußerte. Einer Inspiration folgend schob ich das Papier in den Schlitz an der Seite der Maschine. Dann drückte ich den roten Knopf, nahm die Dokumente, die die Maschine wieder ausspuckte und ging durch das Licht des Deckenstrahlers. Der Raum verschwand, und ich fand mich in einem Sumpf wieder. Das Vorankommen war sehr schwer, da die Stücke festen Bodens nur sehr schmal waren. Gegen die Moskitos kam das Insektenspray zum Einsatz. Am Ende des Sumpfes lag ein Amulett auf dem Boden. Ich kam an ein wunder-





schönes Seeufer, und nachdem ich mich etwas umgeschaut hatte, entdeckte ich ein Seil in einem Loch am Fuße des Baums links im Bild. Ich warf das Seil über einen Ast des Baums und kletterte daran hinauf.

Oben wartete ich eine kleine Weile, und bald erschien ein seltsam gekleideter Mann, der sich auszog und im See badete. Die Gelegenheit ausnutzend, zog ich seine Kleider an. Mein weiterer Weg führte mich nach links, und ich stand vor einem großen Schloß. Mit Erstaunen stellte ich fest, daß hier alles mittelalterlich aussah. Der Wache, die vor dem Schloß stand, zeigte ich das Amulett.

Sie versprach mir, ihrem Herrn davon zu berichten, sobald dieser zurück sei. Um mir die Zeit zu vertreiben, ging ich um das Schloß herum, und bald tat sich eine Lichtung vor mir auf. In der Mitte stand ein großer Baum, über dessen Ast die Kutte eines Mönches hing. Ich schüttelte den Baum, mit dem Erfolg, daß sich eine Münze aus den Falten des Gewandes löste und auf den Boden fiel.

Nach einigem Suchen fand ich sie dann auch. Mit der Münze ging ich in das Wirtshaus. Sie verhalf mir zu einer Mahlzeit, mit der ich einige Zeit überbrücken konnte. Dabei hatte ich Gelegenheit, den Gesprächen am Nachbartisch zuzuhören. Den Worten entnahm ich, daß die Mönche, die hier in einem Kloster sesshaft waren, sich in letzter Zeit mehr als seltsam benahmen. Als ich genug gehört hatte, verließ ich das Wirtshaus und machte mich erneut auf den Weg zur Wache. Nun war der Herr des Hauses anwesend.

Ich trat ein und zeigte ihm das Amulett. Er erzählte mir, daß es seiner Tochter gehöre, die aber verschwunden sei. Leichtsinnig, wie ich nun einmal bin, versprach ich dem alten Mann, seine Tochter zurückzubringen.

Draußen schlief die Wache auf ihrem Posten. Die Gelegenheit ausnutzend, entwand ich ihr ihre Lanze. Mit Hilfe der Lanze konnte ich die Kutte vom Ast holen. Nachdem ich sie angelegt hatte, sah ich aus wie ein waschechter Mönch. Ich ging vor dem Schloß nach Süden und kam an eine Brücke, die von einem großen Wolf bewacht wurde. Ein elektrisches Kabel, das aus seinem Bauch ragte,

brachte mich auf eine Idee: Ich ging zum See und füllte die Tüte mit Wasser. Anschließend begab ich mich auf dem schnellsten Wege wieder zu dem Wolf und warf die volle Tüte über den elektrischen Gesellen, der darauf in einem grellen Blitz verpuffte. Nun war der Weg in das Kloster frei.

Im Kloster gab es einen Raum, in dem sich Mönche ständig rechts herum im Kreis bewegten. Alle Bewegungen, die ich in dieser Halle machte, tat ich also in dieser Richtung und achtete darauf, nicht durch die Mitte des Raumes zu gehen. Mein erstes Ziel war die westliche Tür. Hier gab mir ein Mönch den Auftrag, dem Obervater eine Nachricht zu überbringen. Um nicht aufzufallen, nahm ich diesen Auftrag an. Der Vater befand sich in einem Raum hinter der östlichen Tür. Er verlangte nach einem Becher Wein. Das einzige Behältnis, das sich dazu eignete, befand sich im westlichen Raum. Glücklicherweise war er menschenleer, so daß ich den Kelch an mich nehmen konnte. Hinter der nördlichen Tür befand sich ein Weinkeller, in dem ich den Becher füllen konnte. Mit dem vollem Becher begab ich mich zum Vater. Nach einigen Schlucken war dieser total betrunken und schlief ein. Eine gründliche Untersuchung des Mönchs brachte eine Fernsteuerung zutage, mit der man eine Geheimtür in dem Möbelstück öffnen konnte, hinter der eine Magnetkarte lag. Mit diesen Gegenständen begab ich mich dann wieder in den Weinkeller, wo sich mit der Fernsteuerung das oberste Faß öffnen ließ und einen Weg in ein Labor freigab. Hier war auch die Tochter des Burgherren. Die Karte paßte in den Schlitz ei-



nes Computers, und das hübsche Mädchen wurde auf diese Weise befreit. Das Amulett wies mich als Freund aus. Sie tippte eine Sequenz in den Computer ein und aktivierte die Selbstzerstörung der Anlage und einen Beamer. Kurz darauf befanden wir uns im Schloß, wo mir eine seltsame Geschichte erzählt wurde. Auf jeden Fall stammten die beiden aus der Zukunft und sie wollten mich aus Dank mit dorthin nehmen. Irgend etwas muß aber beim Transport schiefgelaufen sein, denn ich kam allein in einem riesigen Ruinengebiet an. Nach Vollendung dieser Zeilen werde ich mich auf den Weg machen und die Gegend erforschen. Soviel zu Future Wars. Vielleicht können wir in einer der nächsten Ausgaben einen weiteren Teil seines Tagebuches veröffentlichen. Wir bleiben zeitlich gesehen in der Zukunft, denn unsere nächsten Tips, die uns Stefan Stockinger aus Hollabrunn in Österreich zusandte, beziehen sich auf Space Ace, dem Nachfolger von Dragons Lair. Mit Hilfe der Tips sollte es nicht mehr schwierig sein, Commander Borf von seinen teuflischen Plänen abzuhalten.

## Space Ace

Die Tips beschränken sich auf Anweisungen, wann der Joystick in welche Richtung gedrückt werden muß. Zur besseren Orientierung haben wir jeweils eine kurze Beschreibung der Szenen beigefügt. Manchmal ist der Name einer Szene allerdings selbsterklärend.

### 1. Szene "Laserbeschuß":

Space Ace wird von Borf mit einem Laser beschossen. Kurz warten, dann rechts, links, runter

### 2. Szene "Roboter I":

Ein Roboter zerstört den Boden. Sofort links, wenn der rechte Arm oben ist rechts, dann warten, bis der linke Arm runter und wieder rauf ist, links, wenn der linke Arm wieder oben ist links

### 3. Szene "Ufos":

Ace wird von Ufos beschossen. Runter und kurz warten, dann rauf

### 4. Szene "Flug":

Ace landet auf der Station. Kurz vor der Landung: rauf

### 5. Szene "Schlange":

Ace wird von einer auftauchenden Schlange angegriffen: Feuerknopf



### 6. Szene "Stampfer":

Ace steht auf einer Brücke, die von einem Stampfer zerstört wird. Rechts, solange wie möglich warten, dann rauf

### 7. Szene "Dose":

Ace steht vor einer "Dose", die sich auf- und abbewegt. Sobald die Dose ganz unten ist rechts, sofort nach der Landung rechts

### 8. Szene "Monster I":

Ace wird von einem Monster von links angegriffen. Kurz vor dem Monster runter, rechts

### 9. Szene "Monster II":

Ace wird von einem Monster von rechts angegriffen. Kurz vor dem Monster runter, links

### 10. Szene "Monster III":

Ace wird von einem Monster von links angegriffen: Feuerknopf

### 11. Szene "Vor dem Schloß":

Wenn die Katzen am unteren Rand erscheinen, rauf

### 12. Szene "Hunde I":

Ace wird von Hunden verfolgt: rauf

### 13. Szene "Hunde II":

rechts

### 14. Szene "Hunde III":

rauf

### 15. Szene "Roboter II":

Ace steht zwischen zwei Robotern, die auf ihn schießen wollen: Rechts

### 16. Szene "Feuerstrahl I":

links

### 17. Szene "Feuerstrahl II":

links

### 18. Szene "Feuerstrahl III":

rechts

### 19. Szene "Leiter":

rauf

### 20. Szene "Kampf I":

Abwehr, Feuerknopf

### 21. Szene "Kampf II":

Abwehr, Feuerknopf

### 22. Szene "Kampf III":

Abwehr und nach rechts rollen, Feuerknopf, rechts

### 23. Szene "Kampf IV":

Feuerknopf, runter

### 24. Szene "Kampf V":

Abwehr, Feuerknopf

### 25. Szene "Kampf VI":

Ausweichen, rauf, runter

- 26. Szene "Kampf VII":**  
Ausweichen, von hinten anspringen, rechts, runter
- 27. Szene "Huckepack I":**  
keine Aktion
- 28. Szene "Huckepack II":**  
die Naheinstellung, sofort links
- 29. Szene "Plattform":**  
keine Aktion
- 30. Szene "Lavabecken":**  
rechts
- 31. Szene "Schuß I":**  
keine Aktion
- 32. Szene "Einschlag I":**  
dem Schuß ausweichen, rechts
- 33. Szene "Schuß II":**  
keine Aktion
- 34. Szene "Einschlag II":**  
links
- 35. Szene "Schuß III":**  
keine Aktion
- 36. Szene "Einschlag III":**  
rechts
- 37. Szene "Spiegel":**  
Spiegel schieben und zurückspringen, links, rechts - geschafft

Es folgen noch einige Schlußanimationen, dann ist das Spiel zu Ende.

Zak McKracken von Lucasfilm Games erfreut sich nach wie vor einer großen Beliebtheit. Aus diesem Grund haben wir uns entschlossen, den Lösungsweg zu diesem Abenteuer in unsere Helpline aufzunehmen.

## Zak McKracken - Im trauten Heim

In Zaks Wohnung befinden sich eine Menge Gegenstände, die sich später für uns noch einmal als nützlich erweisen könnten. Zunächst wäre da Sushi samt Fischglas, der in unsere Tasche wandert. In der Schrankschublade liegt eine Telefonrechnung, die ebenfalls eingesteckt wird. Apropos Telefon: Um auch während unserer Abwesenheit immer auf dem Laufenden zu sein, schalten wir den Anrufbeantworter ein. Als nächstes fällt uns ein häßlicher Tapetenfetzen an der Wand auf, der abgerissen und mitgenommen wird. Wenden wir uns nun dem Schreibtisch zu. In der Schublade liegt ein Kazoo und unter ihm unsere Cashcard. Pech, daß wir nur schlecht herankommen und sie statt aufzuheben nur tiefer unter den Tisch schieben. Doch mit der Telefonrechnung wird die Aktion zum

Kinderspiel. Soviel zum Thema Schlafzimmer. Unser nächster Weg führt uns ins Wohnzimmer. Das Sofa ist unaufgeräumt - das linke Kissen liegt nicht ordentlich an seinem Platz. Beim Aufräumen stellen wir fest, daß die Fernsteuerung darunter liegt. Hinter dem Sofakissen, das wir soeben von der Wand genommen haben, bemerken wir, daß der Stecker des Fernsehers nicht an seinem Platz steckt: Dem kann man abhelfen. Nachdem nun der Fernseher mit Strom versorgt ist und sich auch die Fernbedienung in unseren Händen befindet, liegt nichts näher, als ihn einzuschalten. Hier können wir zum ersten Mal Annie Larris bewundern. Der Sendung kann man einige wichtige Hinweise auf den weiteren Spielverlauf entnehmen. Weiter geht es in die Küche. Wir bringen das Buttermesser in unseren Besitz. Es hängt über dem Spülbecken. In dem Schränkchen unter dem Spülbecken befindet sich eine Schachtel mit Buntstiften. Auch das Ei aus dem Kühlschrank findet seinen Weg in unsere Tasche. Wenn wir jetzt noch den kleinen Schlüssel neben der Tür an uns nehmen, können wir die Wohnung verlassen.

### Frisch ans Werk

Draußen wenden wir uns nach links und gehen zum Bäcker. Nach dreimaligem Klingeln ist er so verärgert, daß er mit einem steinharten Brot nach uns wirft. Das läßt uns aber kalt, und wir nehmen das Brot an uns. Unser nächstes Ziel ist das TPC-Gebäude, in dem wir uns eine Anmeldung abholen. Sie befindet sich in der Apparatur links hinten an der Wand. Also gehen wir zurück zu unserem Haus, füllen die Anmeldung mit dem gelben Stift aus und legen sie in den Briefkasten, nachdem wir ihn mit dem kleinen Schlüssel geöffnet haben. Als nächstes führt uns unser Weg nach rechts in die 14th Avenue, in der wir das LL-Gebäude betreten. Hier kaufen wir alles, bis auf die Gitarre (den Wet Suit ganz links nicht vergessen!). Weiter nach rechts zum Friseur. Wir öffnen die eben erworbene Werkzeugkiste und schneiden mit dem "wire cutter" das "bobby pin sign" (Haarnadelschild) ab. Nun geht es wieder nach Hause. In der Küche schrauben wir das Abflußrohr mit der "monkey wrench" ab und stecken das Brot in das Spülbecken.



Nach einem Druck auf den Knopf über dem Herd verwandelt sich das Brot in Krumen, die man unten einsammeln kann. Wir verlassen die Wohnung wieder und begeben uns erneut ins TPC-Gebäude, aber nicht ohne uns vorher mit Hut und Brille getarnt zu haben. An dem Computerterminal (erst die Thekentür links aufmachen) können wir dann die Telefonrechnung begleichen (use computer terminal). An dieser Stelle werden wir unseren geliebten Fisch Sushi los, indem wir in liebevoll, aber mitleidslos in den Kübel der Topfpflanze gießen. Das Glas behalten wir aber nach wie vor. Wieder auf der Straße ziehen wir zunächst den Hut und die Brille ab und begeben uns dann zum Bus. Nachdem wir den Busfahrer sanft mit unserem Spiel auf dem Kazoo aus dem Schlaf gerissen haben, öffnet er uns ungern die Tür. Wir treten ein, stecken die Cashcard in den dazugehörigen Reader und fahren in Richtung Flugplatz.

### Wenn einer eine Reise tut...

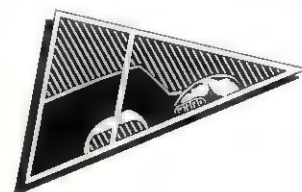
Auf dem Flughafen kaufen wir einem Sektenmitglied ein Buch ab, damit wir auf der lan-

gen Reise auch etwas zu lesen haben. So ausgerüstet können wir das Flugzeug aufsuchen. Um im Flugzeug in Ruhe agieren zu können, müssen wir unbedingt die Stewardess beschäftigen. Dazu gehen wir zunächst ins Heck des Flugzeugs auf die Toilette. Hier stecken wir das Klopapier in den Ausguß des Waschbeckens und drehen den Wasserhahn auf. Wenn das Wasser überläuft, drücken wir die Ruftaste über dem Waschbecken. Jetzt ganz schnell nach rechts zur Mikrowelle und das Ei hineingelegt, die Mikrowelle schließen und anschalten. Von diesem Moment an kann man sich dann wieder Zeit lassen.

Wir untersuchen den ersten Sitzplatz etwas genauer, unter dessen Polster wir ein Feuerzeug finden. Nun untersuchen wir noch die Gepäckfächer. In einem (natürlich dem letzten, egal welches das ist) finden wir eine Sauerstoffmaske, die in unseren Besitz übergeht. Jetzt können wir sowohl den restlichen Flug als auch das Gezeter der Stewardess genießen.

### In Seattle

Nachdem wir den Flug mehr schlecht als recht hinter uns gebracht haben, brechen wir in Seattle einen Ast von einem Baum ab und füttern ein Eichhörnchen mit zwei Köpfen mit den Erdnüssen, die wir von der Stewardess bekommen hatten. Nun können wir mit dem Messer den Dreck von der Felswand abkratzen und legen so den Eingang zu einer Höhle frei. Im nächsten Bild herrscht Finsternis, die man aber mit dem Feuerzeug ein wenig durchbrechen kann. In der Mitte der Höhle befindet sich ein Nest, das wir mit Hilfe des Golfschlägers herunterholen können. Jetzt tasten wir uns zur Feuerstelle voran, in die wir sowohl das Nest als auch den Ast legen. Wenn wir beides mit dem Feuerzeug entzünden, wird es hell und warm in der Höhle, so daß wir uns wesentlich besser orientieren können. Ganz rechts befinden sich seltsame Schriftzeichen an der Wand, die wir mit dem





Stift zu dem Symbol aus unserem Traum ergänzen. Dadurch öffnet sich für uns der Weg in eine Kammer. An den blauen Kristall kommen wir mit Hilfe der Fernsteuerung. Nach getaner Arbeit begeben wir uns zurück zum Flughafen und kaufen uns ein Ticket zurück nach Frisco.

## Wieder zu Hause

Als erstes entnehmen wir dem Briefkasten die King Elvis Fancubkarte (small key benutzen). Ist sie noch nicht angekommen, weil wir zu schnell waren, erledigen wir erst die anderen Dinge hier in San Francisco. Aber wir dürfen sie auf gar keinen Fall hier vergessen! Gehen wir in die 14th Avenue und werfen den blauen Kristall in den Schlitz der roten Tür, worauf uns Annie die Tür öffnet und in die Wohnung bittet. Nun erhalten wir von Annie den ersten Teil des gelben Kristalls und können von nun an zwischen Annie, Zak, Leslie und Melissa hin- und herschalten. Diese Fähigkeit probieren wir auch gleich einmal aus. Wir schalten auf Annie um und nehmen die Cashcard, die unter der Schreibunterlage liegt. Jetzt schalten wir wieder auf Zak und buchen am Flughafen einen Flug über London nach Katmandu.

## In Katmandu

In Katmandu zeigen wir dem Wächter das Buch der Erleuchtung, worauf dieser uns das Haus betreten läßt. Jetzt steht uns der Weg zum Guru offen. Der läßt uns allerdings noch einige Zeit warten, da wir wegen Sushi ein schlechtes Karma haben, das erst noch verfliegen muß. Ist dies geschehen, erklärt uns der Guru, wie wir uns mit dem

gelben Kristall in Tiere verwandeln können. Wieder draußen, zünden wir das Heu mit dem Feuerzeug an. Dies beschäftigt die Polizisten so sehr, daß wir jetzt die Fahne samt Stange vor dem Gefängnis entwenden können. Von dem Yak lassen wir uns für Geld zum Flughafen transportieren. Von hier fliegen wir über London nach Miami.

## In Miami

In Miami tauschen wir mit dem Penner das Buch gegen eine Flasche Whisky und fliegen über Mexico City nach Lima.

## In Lima

Im Dschungel von Lima suchen wir eine Lichtung mit einer Steinplatte. Von jetzt an ist Eile geboten, da wir sonst von einem Alien gefangen genommen werden. Wir legen also die Brotkrumen auf den Stein, warten, bis der Vogel kommt und verwandeln uns in ihn. Als Vogel fliegen wir nach rechts in das linke Auge der Felsfigur, nehmen den Scroll mit und fliegen wieder zurück. Wir übergeben den Scroll an Zak und verwandeln uns wieder zurück. Jetzt nichts wie weg in den Dschungel. Zurück kommen wir jetzt entweder mit dem Flugzeug oder, was wesentlich billiger ist, wir lassen uns bei dem Stein von einem Alien gefangen nehmen, der uns zurück nach San Francisco bringt.

## Die Befreiung

Falls wir also einmal von den Aliens gefangen genommen worden sind, können wir uns ganz einfach wieder befreien. Die Aliens sperren uns in die Verdummungsmaschine und verschwinden. Hier beginnen unsere Befehlscodes langsam

zu verschwinden, also müssen wir uns beeilen. Wir brauchen eigentlich nur den Hut und die Brille aufzusetzen. Nach einiger Zeit kommt dann ein Alien und läßt uns frei, da es uns für einen der ihren hält. Aus dem Schrank können wir dann unsere gesamte Habe an uns nehmen und wieder auf die Straße gehen, wo wir Brille und Hut wieder absetzen. Sollten, aus welchen Gründen auch immer, sich Brille und Hut nicht mehr in unserem Besitz befinden, warten wir einfach, bis wir keine Codes mehr haben und lassen uns von den Aliens auf die Straße setzen, wo die Codes dann langsam wiederkehren. Vom Haus aus kommt man durch einen Geheimgang, der unter einer Bodenplanke beginnt, in das Zimmer mit der Maschine, wo wir unsere Sachen wieder an uns nehmen können.

Wie es mit Zak, Annie, Leslie und Melissa weitergeht, und was sie noch alles erleben, können Sie in unserer nächsten Ausgabe nachlesen. Wir fahren einstweilen mit der Lösung zu einem Sierra-Spiel fort.

## Die Lösung von MANHUNTER NEW YORK

Für alle Menschenjäger hier die Komplettlösung des ersten Teils der neuen Sierra-Serie.

## Der erste Tag

Als erstes werden, wie auch an den folgenden Tagen, alle suspekten Personen mit dem Tracker verfolgt, damit alle Schauplätze erreichbar sind.



Im BELLEVUE HOSPITAL werfen wir einen Blick auf die Leiche und geben ihren Namen ins MAD ein. Der nächste interessante Schauplatz ist die BROOKLYN BAR. Das dortige Videospiel können wir erst nach erfolgreichem Messerwerfen begutachten. Die Bewegung des Mannes nach unserem Erfolg merken wir uns. Beim Videospiel empfiehlt es sich, das Labyrinth im Gedächtnis zu behalten, ebenso sollte es einmal auf dem kürzesten Wege durchlaufen werden. Die Reihenfolge, in der die Figuren dabei abgeschossen werden, gewinnt später noch an Bedeutung. (Siehe auch Karte in AMIGA DOS 2/90.)

Als nächstes suchen wir einen bestimmten Sitz im RESTROOM des PROSPECT PARKs auf. Nach dreimaliger Betätigung der Spülung gelangen wir in die unterirdischen Kloaken. Der Aufbau der Kloake entspricht dem Labyrinth im Videospiel der Bar. An den Stellen, an denen im Spiel Bälle abgeschossen werden, schwimmen Key Cards im Wasser, die wir aufsammeln – insgesamt müssen es zwölf Stück sein. (Es empfiehlt sich, eine Karte des Video-

## SECOND HAND COMPUTER

Ankauf - Verkauf - Vermittlung - Einzahlung

Gebraucht-Computer:

- \* Zubehör
- \* Neugeräte
- \* alle Marken
- \* Konkurrenz-Ankauf
- \* Ankauf defekter Geräte

Angebote solange Vorrat

386er z.B. DELL 30 MHz 8 MB RAM, 300 MB/15 Ms HD, VGA, VGA-Farb-Monitor, Coproz., ESDI-Controller. 15890,-  
IBM z.B. 386-SC, 2 MB RAM, 60 Lb. HD VGA-Monitor/5995,-  
MODEM 2400 Baud, Hayes-kompat., neu, (\*) 399,-  
ATs z.B. 12 MHz, neu, 512 KB, 102 Takt, 200 Watt Netzteil, Monochrom-Karte, Baby-Gehäuse, 1 Laufwerk. 388-SX z.B. 12 MHz, neu, 1 MB, 102 Takt, 200 Watt Netzteil, Monochrom-Karte, Baby-Gehäuse, 1 Laufwerk. 3499,-  
DRUCKER z.B. Mannequin MT81, 130 Z/sk neu. 389,-  
Mannequin MT230, Farbe, Stapel-EBE2 (Neu 5000,-) 2395,-  
(\*) mit (\*) gekennzeichneten Geräten ist der Betrieb in der BRD lt. Par. 15 FAG bei Strafe verboten  
Weitere gebrauchte und Drucker, Monitore, XTs, ATs und 386er, Laptops, Software, RAM-Erweiterungen und vieles mehr auf bei Anfrage!!!

ALPHA 2000 GmbH, 24 Std. Info: 0561-525066, 3501 Niestetal (bei Kassel), Witzenhäuser Str. 10

## JEDES TEIL WOANDERS EINKAUFEN ? BEI SCHAEFER BRAUCHST DU NUR EINMAL ZU LAUFEN!

AMIGA-FLACHBETT-SCANNER Kann als BILDERFASSUNGSGERÄT/Kopierer und Thermodrucker eingesetzt werden. Scannbreite 10 Punkte/Zoll. Scannzeit 10 Sekunden. Abgabe des Bildes im IFF. Aufösungen 640x312 werden unterstützt. Ausschnittvergrößerungen möglich. 10 Graustufen.	DM 949,00
VIDEOTEK-DECODER Kann das VIDEOSIGNAL eines Recorders, Scart-TV oder Tuners den freien Service des Teletextes im IFF- oder ASCII-Format ablegen. Super-Grafik-Darstellung.	DM 298,00
AMIGA-ACTION-REPLAY (Eurosystems) Erweiterungsmodul für AMIGA-500 - Programmierer, Spieltrainer, Zeitlupe, Virus-Detektor, Spritdetektor, Statusanzeige und vieles mehr.	DM 189,00
ZWEITLAUFWERK "Senator 3.5"	DM 259,00
MIDI-MASTER/Midi-Interface (Amiga-Typ) angeben	DM 49,00
MIDI-MUSIC-MANAGER (siehe Eurosystems)	DM 49,00
MIDI-Master und MUSIC-MANAGER	DM 120,00
PRO-SAMPLER und	DM 169,00
GENIUS-MAUS - voll Amiga-kompatibel	DM 79,50
BURSTABLETTER inkl. neuester	DM 149,00
SYNCO EXPRESS inkl. neuester	DM 149,00
- Weiterentwicklung des Burst-Nibbler - Benutzung externer Laufwerke -	
GFA-BASIC 3.0	DM 149,00
DEVPACK-ASSEMBLER (M+T-Verlag)	DM 149,00
AMIGA-DEUTSCH - Deutsche DOS-Befehle einlesen	DM 29,95
SPRACHENKIT - Vokabel-Trainer in Deutsch, Latein, Engl., Französisch, Spanisch, Italienisch, Japanisch	DM 49,95
AMIGA-Ersatzteilepreise wegen starker Schwankungen derzeit nur auf Anfrage	
- Güter aus Markt und Technik - umfangreiche Hardware und Ersatzteile auf Anfrage - Versand nur per Kasse - DM 500 oder Nachnahme -	

## (CLS) - COMPUTERLADEN SCHAEFER

Klingelholl 111, 5600 Wuppertal 2, Tel.: 02 02/50 81 21  
Geschäftszeiten: Mo. + Di., Do. + Fr. 14-18.30 Uhr, Samstags 10-13 Uhr

spiels zu zeichnen und die Positionen der Felder zu nummerieren – dadurch kann man gezielt zwischenspeichern, falls man sich mal verlaufen sollte...) Am Ende des Weges finden wir auf dem Steg ein Medaillon, welches wir aufnehmen.

Das nächste Ziel ist CONEY ISLAND. Dort wählen wir die mittlere Schießbude und schießen die Figuren genau wie im Videospiel ab. Gegen Vorlage des Medaillons erhalten wir eine Data Card, die im MAD folgende Informationen offenbart: der Name 'Phil', offensichtlich ein Verräter, und der Hinweis auf eine ominöse 'Lady', die zerstört werden soll. Die nun erscheinende Frage nach dem Verdächtigen können wir beliebig beantworten; der erste Tag ist somit bereits beendet.

## Der zweite Tag

Im VEND-O-DELI fällt uns auf einem der Pinboards eine Nachricht auf, die von einer 'Lady' handelt und mit 'Harvey' unterzeichnet ist.

Im CENTRAL PARK folgen wir dem Weg der Verdächtigen, indem wir uns anhand gewisser Landmarken (z.B. nach links zwischen einem roten und einem grünen Baum durch) orientieren. An zwei Stellen wuseln die Personen etwas konfus durch die Gegend. Zuerst müssen wir die entsprechende Stelle im Gebüsch finden, wo ein Brecheisen (Crowbar) herumliegt. Haben wir dieses, geht es ein Bild zurück und dann den Tracker-Weg weiter. Am Ende des Weges liegt eine Leiche, von der wir einiges erfahren können. Bei näherem Hinsehen stoßen wir auf die Namen 'Anna' und 'H. Osborne'. Die Kombinationen 'Anna Osborne' und 'Harvey Osborne' lassen unser MAD Erfolg melden. Die Zeichen auf dem Stein scheinen ein

Name zu sein. Mit dem 'P' auf der Stirn der Leiche kommt der Name 'Phil' ins Gedächtnis, und unsere Kombinationsgabe bringt mit 'Phil Cook' einen weiteren Namen ins Spiel, wie das MAD bestätigt.

Der WRETCHED EXCESS NIGHT CLUB ist nur über den Hintereingang zugänglich, der jedoch von vier Schlägern bewacht wird. Die ersten drei gehen ja noch, aber der vierte ist kernig! (Ein Hinweis: an dieser Stelle mit Ton spielen! Es erleichtert die Sache erheblich.) In der Bar sollte auf alle Fälle das Spiel gespeichert werden. Jetzt stehen einige Personen zur Ansicht. Suchen wir uns die richtige aus (ein Tip: Ihre Kutte hebt sich farblich ab), so erhalten wir zwar eines auf die Nase, aber auch eine dreizehnte Key Card und einen Rausschmiß.

In der 152 WEST 82ND STREET, dem Heim von Anne Osborne, finden wir deren Leiche und einen Schlüssel in der Einkaufstasche. Beim MUSEUM öffnen wir die äußere Tür mit dem Schlüssel und folgen dem Weg des Verdächtigen. Die übrigen Türen öffnen wir mit den Key Cards, an der letzten Tür setzen wir das Brecheisen ein. Den daraufhin erscheinenden 'Wachhund' besänftigen wir mit dem Medaillon und gehen weiter. Bei der Leiche im Museum finden sich einige neue Hinweise: Die Zeichnung an der Wand zeigt die Lage eines Gegenstandes in einem Raum. Beim Toten selbst finden wir eine merkwürdige Tätowierung und einen Gegenstand namens Data Module B. Hiermit haben wir das Soll für den zweiten Tag erfüllt und werden wieder abberufen.

## Der dritte Tag

In der TRINITY CHURCH zünden wir die linken Kerzen

in dem Muster an, das auf der Tätowierung zu sehen war. Es öffnet sich ein Geheimfach, in dem das Data Module A sowie einige Symbole zu finden sind. Auf dem FRIEDHOF merken wir uns den Namen 'Jones' auf dem Grabstein.

Im PAWN SHOP erhalten wir einige Badges zur Auswahl, die wir mit den Symbolen aus der Kirche im Hinterkopf auswählen sollten. Nun stehen wir vor einigen Türen, die sich nur nach Lösung einiger Bilderrätsel öffnen. (Die Lösungen sind 41, 1031, 246 und 425.) Am Ende des Ganges finden wir eine Leiche mit dem Namen 'Harry' auf der Kutte. Eine Ecke weiter lauert schon der böse Phil mit seinem Messer. Haben wir ihn in die Flucht geschlagen, hinterläßt er einen Zettel mit einer wichtigen Zahl.

An der Oberfläche testen wir den Namen 'Harry Jones' im MAD. Beim Besuch seiner Wohnung in der PEARL STREET kommt uns die Zeichnung im Museum in Erinnerung und wir schauen uns das Radio genauer an. Mangels Schraubenziehers öffnen wir es mit der Brechstange und nehmen das Data Module C in Empfang. Im THEATER öffnen wir mit den Zahlen des Zettels den Tresor – und finden einen neuen Code, mit dem wir den Computer im EMPIRE STATE BUILDING, der Residenz von Phil Cook, starten können.

Durch ihn erhalten wir wertvolle Informationen über die Organisation der Orbs und können weitere Weichen stellen: Sektor Alpha ist im Krankenhaus angesiedelt. Den dortigen Robot sollte man von 'Special Security' auf 'Hall Patrol' umstellen. Sektor Beta beherbergt die Flotte der Orbs in der Grand Central Station. Sektor Gamma ist in der Statue of Liberty (die 'Lady'?) untergebracht, wie sich zeigt, wenn die dortigen Robots auf Air Defence programmiert werden. (Allerdings sollte man den Schritt rückgängig machen!) Sektor Delta ist das orbsche Rechenzentrum im Empire State Building. Beim Logout werden wir wieder von den Orbs abberufen.

## Der vierte Tag

Heute können wir uns das Tracking sparen, da wir der gesuchte Übeltäter sind. Wenn wir den Robot aus Sektor Alpha richtig program-



miert haben, können wir das Krankenhaus genauer erkunden. Zwar werden wir vom Wächter gefangen, ein Fenster, mit der Brechstange zu öffnen, bringt jedoch interessante Einblicke. Nachdem die Orbs den Raum verlassen haben, können wir der Maschine das Data Module D entnehmen und über das eingeschaltete Fließband das Krankenhaus wieder verlassen. In die GRAND CENTRAL STATION gelangen wir mit dem Brecheisen. Dort wird das Raumschiff mit den Data Modules komplettiert und gestartet. (Betätigung von Schalter links, Taste Mitte, Taste links, Taste rechts.)

Durch Schleuse und Labyrinth gelangen wir zum Startplatz. Nach Betätigen des rechten Schalters und dem Blick auf die Anzeige beginnt der letzte Luftkampf. Die vier Sektoren der Orbs müssen zerbombt werden, wobei man sich jedoch auch vor Phil in acht nehmen muß. Sind Freiheitsstatue, Bellevue Hospital, Grand Central Station und Empire State Building zerstört (erfolgreiche Abwürfe werden mit dem Bild der zerstörten Örtlichkeit quittiert!), ist das Spiel gewonnen.

Damit haben wir das Ende der Helpline erreicht. Wir hoffen, daß der eine oder andere Tip bei einem Ihrer Spielprobleme helfen konnte.

(M. Anton/R.Marz/hs)





# Wir haben alles für Ihren Amiga!

## Neues:

	DM
688 Attack Sub	74,-
Apprentice	54,-
Bat	79,-
Black Tiger	74,-
Budokan	69,-
Castle Master	69,-
Dragon's Breath	84,-
Eagle Rider	69,-
Escape from Singe's Castle	119,-
(Dragon's Lair II)(512kb)	
First Contact	69,-
Gold of the Americas	69,-
Infestation (Psygnosis)	74,-
Killing Game Show	54,-
Logo	74,-
Midwinter	79,-
North Sea Inferno	39,-
Paris - Dakar 90	69,-
Pipe Mania	74,-
Survival	69,-
Their finest Hour	79,-
Tie Break	74,-
Tower of Babel	69,-
Window Wizard	49,-

## Die Hitliste:

	DM
Xenon II	74,-
Cabal	69,-
Rings of Medusa	74,-
Indiana Jones Abenteuer	69,-
Populous	69,-
It came from the desert	84,-
Drakkhen	84,-
Sim City	89,-
North & South	59,-
3D-Blockout	69,-
Kick Off	49,-
Starflight	69,-
Kaiser	119,-
Supercars	69,-
Rainbow Island	69,-
Test Drive II	74,-
Future Wars	69,-
E.S.S.	84,-
Conqueror	69,-
Austerlitz	74,-
Great Courts	69,-
Hard Drivin'	59,-
Iron Lord	69,-
Extra Time (Kick Off erf.)	34,-

## Klassiker:

	DM
Airborne Ranger	64,-
Aquanaut	74,-
Axels Magic Hammer	59,-
Balance of Power 1990	79,-
Battlehawks 1942	64,-
Battletech	79,-
Beach Volley	74,-
Bloodwych	74,-
Börsenfieber	69,-
Bundesliga Manager	59,-
Chambers of Shaolin	74,-
Die Stadt der Löwen	104,-
Dragon's Lair	104,-
Dungeon Master	74,-
Dyter 07	49,-
Earl Weaver Baseball	29,-
F16 Combat Pilot	69,-
Jack Nicklas Golf	74,-
Shot em'up Construction Set	89,-
Stunt Car Racer	69,-
Super Wonderboy	74,-
Wayne Gretzky Hockey	69,-
Weird Dreams	69,-
Wild Street	74,-

## Ankündigungen:

Bitte beachten Sie, daß diese Programme bei Drucklegung noch nicht lieferbar waren. Bitte erfragen Sie Preis u. Verfügbarkeit telefonisch.

Pop-Up  
Bobbie Plus  
Clax  
Xenomorph  
Tennis Cup  
Cyberball  
Okolopoly  
Leisure suit Larry III  
Player Manager  
Champions of Krynn  
Lost Dutchman Mine  
Sherman M4  
Turrican  
Startrash  
Chaos strikes back  
Kings Quest IV  
Hero's Quest  
East vs. West

# Super!

Amiga-Spiele  
ab DM

# 9.--

(nicht über Versand!)

Laden

**Peksoft**

Versand

Müllerstr. 44  
8000 München 5

Mo-Fr 9-13 und 14-18 Uhr  
Samstag 9-13 Uhr



089/260 93 80

Landsberger Str. 77  
8031 Gilching

Fax: 08105/231 09



08105/8037

Versand per NN+DM8,- oder Vorkasse/Kreditkarte+DM6,-. Ab DM 350,- Bestellwert portofrei und verpackungsfrei. Ausland nur gegen Vorkasse/Kreditkarte +DM10,-



DINERS/EURO/VISA

# Über 250

Public Domain  
Disketten im  
Laden!!!

(nicht über Versand!)

## Zubehör:

Diskettenlaufwerk 3.5" anschlussfertig  
als 2.Laufwerk für Amiga 500 DM 249,-

Diskettenlaufwerk 5.25" anschlussfertig  
als 2.Laufwerk für Amiga 500 DM 298,-

Speichererweiterung für Amiga 500 auf  
1MB inklusive Spiel Dungeon Master DM 249,-

Speichererweiterung abschaltbar mit  
Uhr inklusive Spiel Dungeon Master DM 269,-

Disketten 3.5" 2D No Name  
mit Garantie. 10er Packung DM 14,90

TDK Disketten 3.5" 2D  
mit Garantie. 10er Packung DM 34,90

Disketten 5.25" 2D/HD No Name  
mit Garantie. 10er Packung DM 19,90

## Vom Stadtjugendamt München empfohlen:

	DM		DM
Rainbow Warrior	74,-	Lombard RAC Rally	74,-
Airball	54,-	Mini Golf	59,-
Bubble Ghost	59,-	Fugger	59,-
Pacmania	59,-	Elite	59,-
Ferrari Formula One	79,-	Oil Imperium	54,-
Test Drive II	74,-	Chessmaster	69,-
Tetris	49,-	Turn It	54,-
Skweek	54,-	Espionage	59,-
Ballistix	54,-	Hollywood Hijinx	89,-
Powerdrome	64,-	The Pawn	69,-
Speedball	79,-	Guild of Thieves	69,-
Clown-O-Mania	54,-	Times of Lore	69,-
Rock'n'Roll	64,-	Bureaucrazy	69,-
Circus Games	79,-	Zak McKracken	64,-
The Games - Summer Edition	74,-	Balance of Power 1990	69,-
Microprose Soccer	69,-	Kaiser	119,-
Volleyball Simulator	49,-	Wall Street Wizard	69,-
Wayne Gretzky Hockey	69,-	Space Quest	74,-
Great Courts	69,-	Asterix im Morgenland	59,-
RVF Honda	69,-	Shanghai	69,-

Liebe Eltern!

Sind Sie sich nicht sicher, welches Spiel das Richtige für Ihren Sprössling ist? Kommen Sie doch ganz einfach in unseren Laden und lassen Sie sich beraten!



Unheimlich ist es im finsternen Wald. Was wird Sie hier noch alles erwarten?

## Dungeon Quest

Ausgeraubt, ohne Waffen und Geld, befinden Sie sich mitten in der Wildnis und wissen nicht mehr weiter. Dies haben Sie einem Teil des Hilferufes zu verdanken, der Sie per Post erreicht hat. Absender war einer Ihrer besten Freunde, und er scheint wirklich in großen Schwierigkeiten zu stecken...

**D**ungeon Quest ist ein Textadventure, sucht man Icons oder ähnliches vergebens. Zu den Texten kommen noch gute Grafiken und einige Soundeffekte, die zwar stereo, aber doch eher nervend sind. Führt man einen Ortswechsel durch, so wird man mit einer neuen Grafik belohnt, die der Fantasie auf die Sprünge helfen soll.

Der Parser gehört zwar nicht gerade der Elite an, aber man kann ganz gut mit ihm arbeiten. Er weiß auf fast jede Eingabe eine einigermaßen vernünftige Antwort. Allerdings weist er einige Eigenarten auf: Will man zum Beispiel etwas untersuchen, erhält man auf die Eingabe "EXAMINE" meistens die Antwort "Your search turns up to nothing", "EXAMINE" ohne weitere Zusätze hingegen listet sämtliche Gegenstände in Form von Texten auf.

Die Story ist im Fantasy-Bereich angeordnet und relativ interessant. Etliche bekannte Elemente, wie der skelettierte Fährmann mit dem einzigen Schiff über den unheimlichen See oder hungrige Drachen, sind enthalten. Obwohl bis an den Rand mit Klischees gefüllt, gelingt es der Story doch, einige Spannung und

damit Spielmotivation zu erzeugen. Abschließend ist zu sagen, daß Dungeon Quest ein Text-Gratik-Adventure der Mittelklasse ist. Im mittleren Bereich ist auch der Schwierigkeitsgrad anzusiedeln. (Robert Marz/hs)

### AMIGA DOS Blitzlicht

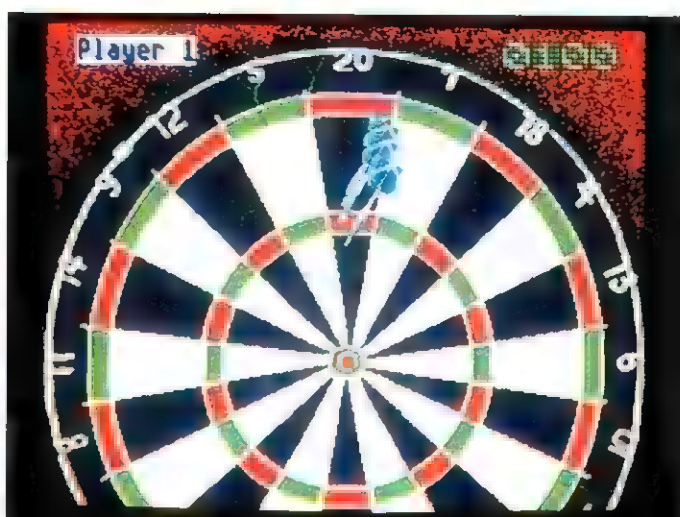
**Name:** Dungeon Quest  
**Hersteller:** Gainstar  
**Quelle:** Fachhandel  
**Preis:** zirka 85 DM

#### Positiv:

- Spielatmosphäre
- gute Grafiken
- Anleitung deutsch

#### Negativ:

- Spiel komplett in Englisch
- schwache Anleitung



Eine ruhige Hand und große Treffsicherheit sind vonnöten, um in den Wettkämpfen bestehen zu können

## John Lowes Ultimate Darts

Darts gehört neben Billard zu den beliebtesten Kneipensportarten. Kein Wunder also, daß es auch auf den Heimcomputern Einzug gehalten hat. John Lowes bietet neben der eigentlichen Simulation auch noch einen Modus, in dem es als Anzeigetafel für eigene Wettbewerbe dient.

**I**n Ultimate Darts wurden verschiedene Spielvarianten vereinigt. Da gibt es normale Wettbewerbe, an denen bis zu acht Gegner teilnehmen können, und Spiele wie Rund um die Zielscheibe, Darts Fußball und Darts Cricket.

Im Arcademodus bewegt sich der Pfeil, von der Maus gesteuert, in kreisenden Bewegungen über die Zielscheibe. Wenn man dann klickt, wird der Pfeil nach einer kleinen Verzögerung geworfen. Trifft man einen Pfeil, der schon auf der Scheibe steckt, so prallt der neue Pfeil ab und fällt zu Boden. Es gibt auch eine Option Autowurf, bei der der Computer nach einer bestimmten Zeit automatisch wirft, auch wenn man mit dem Zielen noch nicht fertig ist. Schließlich gibt es noch eine Option mit dem Namen Schaukampf. Hier kann man gegen die besten Spieler der Welt antreten, deren spezielles Wurf- und Treffverhalten in den Computer eingegeben wurde.

Darts-Freunde werden ihren Spaß haben und auch den Realmodus für echte Wettkämpfe zu schätzen wissen. Wer diesem Spiel allerdings schon in

der Realität keinen Spaß abgewinnen kann, der wird auch durch dieses Programm nicht zum Darts-Fan.

(Robert Marz/hs)

### AMIGA DOS Blitzlicht

**Name:** John Lowes Ultimate Darts  
**Hersteller:** Gremlin  
**Quelle:** Fachhandel  
**Preis:** zirka 65 DM

#### Positiv:

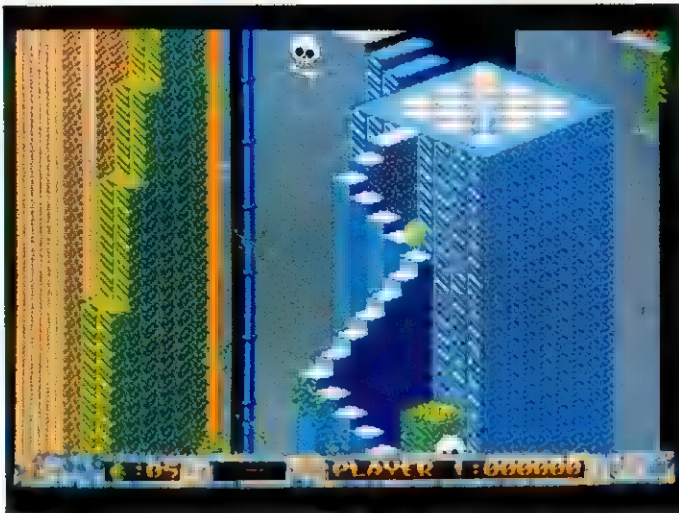
- Realmodus Anzeigetafel

#### Negativ:

- nur für Darts-Freunde interessant







Viele Abenteuer muß der kleine Neuroflummi bei seiner Aufgabe, die Akten und Formulare der Galaxis zu retten, bestehen

## Star Trash

Wer kennt sie nicht, die Spiele, in denen eine kleine Kugel über schmale Grate von Gebirgen gesteuert werden muß? Bei Star Trash ist die Kugel ein Neuroflummi und muß die verlorengegangenen Akten der galaktischen Verwaltung von einem Gebirgsplaneten retten.

**D**er Neuroflummi springt und rollt, von Ihnen gelenkt, durch die Landschaft, in der er die verlorenen Akten suchen muß. Sie können zwischen drei verschiedenen Möglichkeiten der Steuerung wählen.

Der Neuroflummi springt also durch die Gebirgslandschaft und sammelt Gegenstände, die er findet, auf. Da wären zum Beispiel Bonbons und die Akten, die Punkte bringen, oder Blubb, der Fisch, der ebenfalls Punkte bringt, aber schon gefundene Schlüssel aufruft. Es gibt allerdings auch Wesen, die dem Flummi schaden wollen: Staubsauger, die ihn einsaugen, oder riesige Seepferde oder Dämonen, die buchstäblich auf ihm herumtrampeln. Kommt der Flummi also mit einem solchen unfreundlichen Lebewesen in Berührung oder stürzt er in Folge eines Fehlsprunges in einen Abgrund, verliert er Punkte oder eines seiner anfänglich fünf Leben.

Star Trash kann mit bis zu vier Spielern gespielt werden und bereitet gerade durch seine vielen Animationen sehr viel Spaß. Durch immer neue Gags und Extras bleibt die Motivation über lange Zeit erhalten. Mit Star Trash liegt mal wieder ein Geschicklichkeitsspiel der freundlichen Sorte vor.

Friedlich, niedlich und spaßig. Auch wenn an keiner Stelle der Oscar für besondere Leistungen vergeben werden kann, ist ein solches Spiel inmitten blutrünstiger Aliens und bewaffneter Telespielkonflikte eine echte Wohltat.

(Robert Marz/hs)

### AMIGA DOS Blitzlicht

**Name:** Star Trash  
**Hersteller:** Rainbow Arts  
**Quelle:** Fachhandel  
**Preis:** 59,95 DM

#### Positiv:

- viel Spaß
- hübsche Animationen
- verschiedene Steuerungstypen
- Zeitlupenoption zum genauen Springen

#### Negativ:

- keine Speicheroption



Flugabenteuer über der bezaubernden Inselwelt von Hawaii - Szenario-Disketten machen es möglich

## Hawaiian Odyssey

Dienstagabend ist Krimizeit, pünktlich ab 21.45 Uhr beobachte ich die wilden Verfolgungsjagden, die Detektiv Magnum mit seinem Freund T.C. im Hubschrauber rund um Hawaii unternimmt. Ach, wie gerne würde ich mal mitfliegen! Und siehe da, es geht, und zwar mit der Scenery-Disk namens Hawaiian Odyssey!

**D**er Flight Simulator hat sich nicht nur auf dem PC als die Flugsimulation überhaupt entpuppt, sondern auch auf dem Amiga. Aber immer nur über der Bucht von San Francisco ins Trudeln zu kommen, ist auf die Dauer langweilig. Die neue Scenery-Disk bietet da einiges an Abwechslung, man kann von jeder der sechs Hauptinseln Hawaiis einen Flug unternehmen.

Das Fliegen selbst gestaltet sich wie gehabt, daß der FS-II kein 'Drauflos-Flieger' ist, versteht sich von selbst. Ausgestattet ist das Softwarepaket mit einer Übersichtskarte von Hawaii mit den wichtigsten Flugplätzen, den Koordinaten der Flugplätze und den Radiofrequenzen.

Die Flugszenen entbehren nicht eines gewissen Reizes. Auch wenn die Vektorgrafik immer zu glatt ist und eben Simulation bleibt, bildet sich beim Flug über den Krater eines aktiven Vulkans schon ein mulmiges Gefühl in der Magengegend.

Die 'Hawaiian Odyssey' ist ein Muß für alle FS-II-Besitzer, nicht nur, weil ohne das Hauptprogramm nichts läuft (oder fliegt), sondern auch

wegen der vielen neuen Situationen, vor die der 'Heimflieger' gestellt wird. Auf denn, Burschen, die Hawaii-Sonne geputzt...

(jb)

### AMIGA DOS Blitzlicht

**Name:** Hawaiian Odyssey, Scenery-Disk für Flight-Simulator II  
**Hersteller:** Sub Logic  
**Quelle:** Fachhandel  
**Preis:** 59,95 DM

#### Positiv:

- realistische Darstellung der Hawaii-Inseln und Flughäfen

#### Negativ:

- teilweise grobe Vektorgrafik





Footballer of the Year – von Fußball kaum eine Spur!



The Cycles: Rasen, bis der Motor qualmt...

## Footballer of the Year 2

Sport ist Mord, hier ist ein Beispiel dafür, wie man mit Sportspielen den Ruf eines Computers und die Nerven des Spielers ruinieren kann...

Das Ziel ist hoch: man möchte Fußballer des Jahres werden. Zunächst sucht man sich einen passenden Verein aus den englischen Ligen, danach geht es der Kugel ans Leder – von wegen... Denn hier steht weniger der Sport als vielmehr das Geld im Vordergrund. Davon braucht man nämlich jede Menge, wenn man kicken will! Nur wer sich 'Torkarten' leisten kann, darf auch versuchen, Tore zu schießen. Vor jedem Spiel können ein bis drei dieser Karten, die jeweils mit einem bestimmten Manöver verbunden sind, eingesetzt werden. Die Manöver werden dann auf einer Tafel gezeigt und müssen vor dem gegnerischen Tor in der Vogelperspektive nachgespielt werden. Mit etwas Glück (oder Training) erzielt man dabei tatsächlich ein Tor (und etwas Geld), den Rest der Zeit verbringt man mit Bewunderung von Statistiken oder dem Warten auf zufällige Ereignisse wie Auf- oder Abstiegsmöglichkeiten. Eine gar seltsame Art von Fußball... Man hat so den Eindruck, der einzige Zweck des Programms ist es, einen sensationell langsamen Algorithmus zum Zeichnen von Linien zu präsentieren. Einen Vorteil hat das Spiel

vielleicht doch: Man kann sich beim Gerangel um den Ball nicht verletzen. Dafür besteht jedoch ernsthafte Verletzungsgefahr, wenn man sich nach dem Kauf und erstem Probespiel 'sonstwohin' treten möchte...

(Michael Anton/hs)

### AMIGA DOS Blitzlicht

**Name:** Footballer of the Year  
**Hersteller:** Gremlin  
**Quelle:** Fachhandel  
**Preis:** zirka 64 DM

#### Positiv:

- die Anleitung eignet sich sehr gut zur Stabilisierung wackeliger Tische

#### Negativ:

- dafür ist der Preis jedoch zu hoch...



## The Cycles

Wer mit den bislang für den Amiga erschienenen Motorradsimulationen noch nicht genug hat, kann sich jetzt mit The Cycles erneut auf den Balanceakt mit zwei Rädern begeben.

Das Rezept für die Entstehung von The Cycles war wohl folgendes: Man nehme ein Auto aus Test Drive II, halbiere es und gestalte die Fahrtstrecken in der Form eines Kreises. Tatsächlich erinnert vieles an dieses legendäre Straßenrennen, vor allem kleine Details wie der wählbare Schwierigkeitsgrad oder die Qualmwolke im (für ein Motorrad sehr unlogisch platzierten) Rückspiegel beim Überdrehen des Motors. Dies ist auch nicht verwunderlich, da für beide Spiele Accolade und Distinctive Software verantwortlich zeichnen. Allerdings geht es bei The Cycles nicht um geschicktes Ausmanövrieren von Cops, sondern um möglichst gute Rundenzeiten. Die müssen gegen eine Vielzahl von Mitfahrern auf weltberühmten Rennstrecken in drei Hubraumklassen erkämpft werden.

Die Gestaltung des Rennens ist solider Durchschnitt, der zu Vergleichen mit anderen Programmen dieser Art reizt. Im Kontrast zu Super Hang On ist der Sound relativ mager, die grafische Gestaltung ist nicht ganz so liebevoll wie dort, aber wesentlich besser als in RVF Honda. Die von den Herstellern hervorgehobene einmalige 'Ego-Perspektive' ist jedoch nicht sonderlich

sensationell, da bei The Cycles die Kamera lediglich etwas nach vorne gesetzt wurde. Der Sound beschränkt sich auf die Titelmusik und Motorengeräusche, die Steuerung ist akzeptabel. Der ultimative Fahr-simulator ist The Cycles wohl nicht.

(Michael Anton/hs)

### AMIGA DOS Blitzlicht

**Name:** The Cycles  
**Hersteller:** Accolade  
**Quelle:** Fachhandel  
**Preis:** 84,95 DM

#### Positiv:

- relativ gute Grafik
- wählbare Schwierigkeit

#### Negativ:

- nichts wirklich Neues





# Gewinnen Sie mit Magic Bytes!

Nicht nur Computerspiele machen Spaß, dachten sich die Macher bei Magic Bytes und unsere Spielere-redaktion. Warum nicht einmal ein richtig tolles Preisrätsel veranstalten, bei dem es auch etwas Tolles zu gewinnen gibt?

**G**esagt, getan. Rund um den Bohrrinselthriller 'North Sea Inferno', das neue, spannungsgeladene Magic-Bytes-Amiga-Spiel, haben wir uns ein kleines Bilderrätsel ausgedacht, das es zu knacken gilt. Wir haben auf dieser Seite drei verschiedene Bildschirmfotos zum Spiel 'North Sea Inferno' abgedruckt. Eines dieser drei Bilder stammt nicht aus diesem Spiel. Ihre Aufgabe ist es, herauszubekommen, welches der Bilder nicht paßt. Wenn Sie das richtige Foto gefunden haben, schreiben Sie dessen Nummer und Ihre Adresse auf eine Postkarte.

Unter den richtigen Einsendungen werden folgende Preise verlost:

1. Preis: 1 Taschenfernseher mit LCD-Flachbildschirm.
2. Preis: 3 Magic-Bytes-Spiele nach Wahl.
3. bis 5. Preis: Je ein Magic-Bytes-Spiel nach Wahl.

Ihre vollständig ausgefüllte Teilnahmekarte schicken Sie bitte an folgende Adresse:

DMV-Verlag  
Red. AMIGA DOS  
Kennwort Nordsee  
Postfach 250  
3440 Eschwege



Der Einsendeschluß ist der 08.05.1990, es gilt das Datum des Poststempels. Mitarbeiter des DMV-Verlags und deren

Angehörige sind von der Teilnahme ausgeschlossen. Der Rechtsweg ist ausgeschlossen. (hs)

AMIGA - Hits zu TOP-Preisen!!!	
Blasteroids Der Hit von der Spielhalle... 29.90	Ninja Warriors M. Anleitung, viel Action, spitzen Grafik... 29.90
Spy vs Spy Mit 2. Anleitung und doppeltem Screen... 29.90	Fright Night Der Vampire besitzt du für nur... 19.90
Quantum Arcade-Action per mit 32 Angriffswellen... 19.90	Tarzan Adventure-Action mit 120 Landschaften, 40 verschiedenen Charakteren und digitalisierten Sound... 29.90
Starkey Von Logotron, mit Super-Scrolling, 31 Angriffswellen in 4 Welten und digitalisierten Sound... 29.90	See Sea Air Rally Simulation und Action auf Top-Grat... 29.90
Nigel Mansell GRAND PRIX Mit Turbo Speed und 3-D um die Ehren... 29.90	Phantom Fighter Der Arcade-Schoner zum Top-Preis... 19.90
Tetra Quasta Die ersten galaktischen Spiele jetzt nur... 19.90	Star Wars-Trilogy Star Wars Teil 1, 2 und 3 zusammen nur... 39.90
Black Shadow von Electronic Arts mit 16 Action... 29.90	Boulderdash Der Klassiker zum fantastischen Preis... 29.90
Incredible Shrinking Sphere Der Hammer von Electronic Dreams: 138... 29.90	Strippoker 2 PLUS New Girls und New Tracks... 29.90

## PROGRAMME!

### AKTUELLE HITS

Nordic Power	PREZIER 189.90
7 Gates of Jambala	79.90
Airborne Banger	49.90
Aqueduct	79.90
Austerlitz	79.90
Bad Company	59.90
Battle Squadron	59.90
Block Out	79.90
Bloodwych	59.90
Blue Angel	59.90
S. Illiger Supersoccer	79.90
Boruma	59.90
Chase HQ	59.90
Darius	79.90
Dark Fusion	59.90
Day of the Viper	79.90
Drekhen	89.90
Dr. Doom's Revenge	79.90
Dungeon Quest	79.90
P. Boardsley Eurosoccer	29.90
Extra Time	29.90
Fifth Gear	59.90
Fighter Bomber	89.90
Football 2 Pack	49.90
Footballer o.th.V. 2	59.90
Full Metal Planets	79.90
Future Wars	59.90
Ghost n' Ghouls	79.90
Sole in One Golf	79.90
Iron Lord	79.90
It Came fr. the Desert	59.90
Future Wars	79.90
Krypton Egg	59.90
Luxes & Napoleon	59.90
Lösung Indiana Jones	17.90
Musik Mansion 41	79.90
Mark II	69.90
Windbender	59.90

Neu  
Der Arcade-Hit für  
den Amiga... 39.90  
Hard Drivin  
39.90

### Public-Domain-Software

Wir haben über 2000!!! Disketten (fast) allen Gebieten und PD-Serien, die wir ab DM4.90 weitergeben. Inhaltsverzeichnis und Kurzbeschreibungen auf:  
2 Disketten für AMIGA... 10.00

### AKTUELLE HITS

Ninos	59.90	Supervonderboy	79.90
Omega	59.90	Tablennis	59.90
Goza	79.90	Testdriven Muscle Cars	39.90
Outlands	29.90	The Beast	39.90
Gafari Guss	59.90	The Cyclone	79.90
Scrabble Spades	59.90	Total Eclipse	79.90
Soccer Manager 2	49.90	Turo It	79.90
Space Ace	129.90	TV Sports Basketball	69.90
Space Warrior 2	59.90	Ultimate Darts	59.90
Spherical	59.90	Untouchables	79.90
Star Blazer	59.90	Vatp	59.90
Star Wars	29.90	Waterloo	79.90
Stargos	59.90	Wild Streets	79.90
Supercars	59.90	X Copy + Hardware	59.90

**SOFORT-BESTELLUNG  
PER TELEFON  
09 11 / 28 82 86**

### TOP-Adventures für AMIGA!!!

PIEM Das Original von Fantasy von Preis von... 39.90
Waterloo von P2P: die Total-Simulation mit riesigen Bandwidth, großer Spieltrieb und 3-D-Grat... 39.90
Shadowgate Das Abenteuer von Hadescape mit 100%+Taste 39.90
Cartier-Command 3-D-Simulation mit Action und Spieltrieb... 39.90
BISMARCK Original von P2P: Bismarck Action nur... 39.90

### TOP-Adventures für AMIGA!!!

Strike Force Harrier Der legendäre Simulator von Microsoft... 29.90
Dark Castle Der Side-and-Death Aft. mit Grafik & Sound... 29.90
Armageddon Ren Bestimmen Sie über die Welt in Jahr 2012: mit Voll-Systemen (Editor, Platten und 16 Screens)... 29.90

Alle Preise sind unsere Ladenpreise. Bei Versand berechnen wir anteilige Selbstkosten: bei Vorkasse mit Scheck: DM 2,50, bei Versand per Nachnahme DM 5,90 je Sendung.

**T.S. Datensysteme**

DENISSTRASSE 45 8500 NÜRNBERG 80 TELEFON 0911/288286

### BESTELLUNG + INFO ANFORDERUNG

☐ Hiermit bestelle ich für den Computer

☐ Nachnahme (+ Kosten 5,90)

☐ Vorkasse und Check (+ Kosten 2,50)

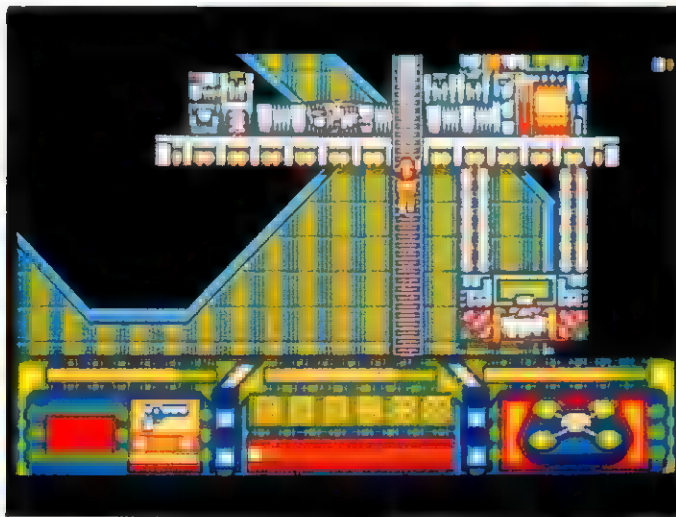
☐ Ich möchte ein kostenloses Gesamtinfo über Software für meinen Computer.

Bitte Anschrift nicht vergessen \_\_\_\_\_ Unterschrift  
AMIGA DOS 5/90  
T.S. Datensysteme · Denisstraße 45 · 8500 Nürnberg

**D**ie Vorgeschichte zu Stryx ist zwar nicht gerade Pulitzerpreisverdächtig, hebt sich jedoch etwas von den üblichen Legitimationen eines Gemetzels unter 'Un-Menschen' ab. In recht ferner Zukunft bedrohen ausnahmsweise keine Aliens die Menschheit, vielmehr wenden sich die Ergebnisse der modernen Zivilisation, hochgezüchtete Androiden, gegen ihre Erzeuger. Wie es dazu kam oder kommen wird? – Eine gute Frage...

Wir schreiben das Jahr 3106, welches das Ende eines langen Krieges einläutet, der die Menschheit an den Rand des Ruins brachte. Das Leben ist nur noch in der Domstadt möglich, einem Zusammenschluß von fünf miteinander verbundenen 'Käseglocken'. Die Zivilisation stabilisiert sich wieder, im Laufe der Zeit erkennt man auch die Ursachen für den verheerenden Krieg: Eigentlich waren nur defekte Computer und Roboter am Chaos schuld. Also führt man ein Zeitlimit ein, nach dessen Überschreitung alle Computer und Roboter zerstört und durch neue ersetzt werden, bevor ihre künstliche Intelligenz zu gefährlich werden kann. Diese Aufgabe wird der 'Lifeforce', einem Kontrollmechanismus, anvertraut, der nur durch vier Schlüssel aktiviert werden konnte. Dieses System läuft einige Jahrhunderte lang recht gut, im Jahre 4516 kommt es jedoch erneut zu einer Rebellion der Roboter. Diese klauen die Schlüssel zur 'Lifeforce' und beginnen, alles Leben in der Domstadt zu zerstören...

Klar, daß hier ein erhabener Held helfen muß, ebenso klar, daß Sie als Spieler herhalten müssen. Der Retter heißt Stryx



Für Stryx sind die Roboter Freiwild – und umgekehrt!



**In Stryx darf wieder zum Wohle und zur Rettung der Menschheit geballert werden. Diesmal jedoch, um ein selbst eingebrocktes Süppchen auszulöffeln, das mit einer Horde wildgewordener Androiden gewürzt ist...**

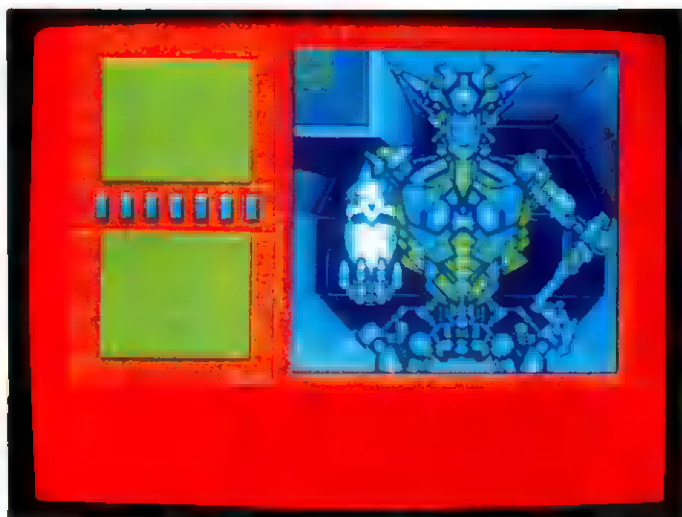
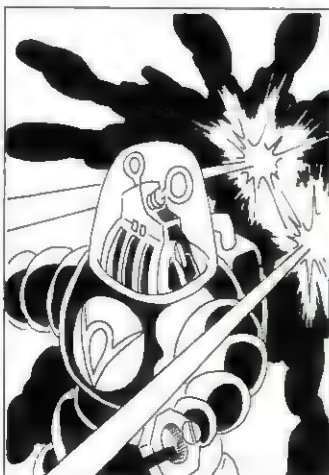
und ist halb Mensch und halb Roboter, 'the meanest fighting, smartest thinking machine ever invented'. Also den Ballermann geschultert und auf in den Kampf. Fünf Dome müssen nach den Schlüsseln für 'Lifeforce' und andere Türen durchsucht werden, in jedem Dom warten viele Roboter auf ihre Terminierung. Um

von einem Dom in den anderen zu gelangen, müssen Kanäle durchquert werden, in denen es von Gegnern nur so wimmelt. Hilfsmittel gibt es genug – wenn man rechtzeitig an sie herankommt. In den Domen sind Waffen, Jetpacks und Energiereserven zu finden, in den 'Hives' unter den Domen kann man sich zusätz-

lich mit Power versorgen, allerdings geht es auch hier heiß her. Hat Stryx alle Schlüssel gefunden, braucht nur noch die 'Lifeforce' aktiviert zu werden, und der Spuk hat ein Ende...

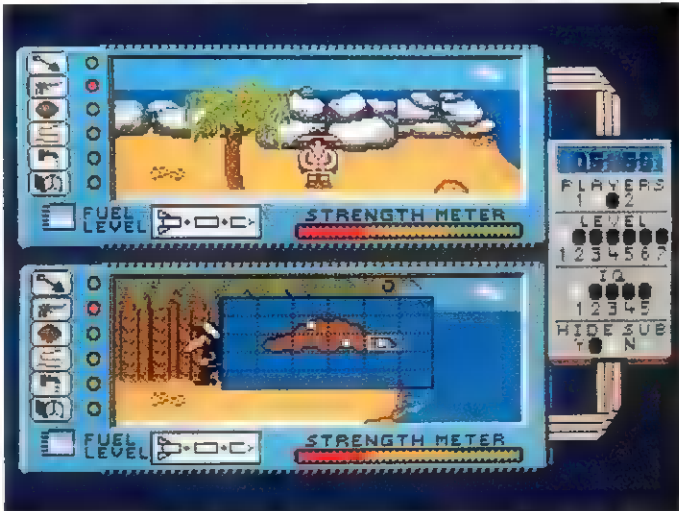
Dies ist jedoch leichter gesagt als getan, denn der Schwierigkeitsgrad des Spiels ist irgendwo zwischen unmöglich und unfair. Dies beginnt mit der Vielzahl der Gegner, die den Helden mit Vorliebe von den Plattformen pusten (Strategie und Reaktion sind gefragt) und endet mit fehlenden Save- und Continue-Optionen und dem (sehr schnell verbrauchten) einzigen Bildschirmleben. Ob dies motiviert oder nicht ist reine Geschmackssache – die Vielfalt der Aufgaben und Schauplätze besitzt jedoch ihren Reiz. Vom technischen Standpunkt spricht jedenfalls wenig gegen das Spiel: Die Titelsequenz ist sehens- und hörens- wert, der Sound im Spiel etwas dürrig, die Grafik insgesamt gelungen – lediglich die Steuerung in den 'Hives' vermag nicht zu überzeugen. Alles in allem verspricht Stryx viele Stunden Ballerspaß – vor allem, wenn man etwas masochistisch veranlagt ist. (Ich wünsche mir nichts sehnlicher, als einen soliden Cheat-Mode...)

(Michael Anton/hs)



AMIGA DOS Blitzlicht		
Name:	Stryx	
Hersteller:	Psychapse	
Quelle:	Fachhandel	
Preis:	zirka 70 DM	
Positiv:	- viel Abwechslung	
Negativ:	- sehr schwer!	
GRAFIK	SOUND	MOTIVATION
7	6	8





Die MAD-Haus- und -Hofspione beim Einsatz auf einer einsamen Insel

## SPY VS THE ISLAND SPY™

In dem satirischen Monatsmagazin MAD sind die beiden schon zur Institution geworden. Monat für Monat gehen der schwarze und der weiße Spion auf die Pirsch, um einander eins auszuwischen.

Irgendwann einmal wurde aus den beiden Spionen auch ein Computerspiel. Zur hohen Zeit des C-64 durften die beiden zwei Abenteuer bestehen. Nun liegt der leicht eingestaubte zweite Anlauf auch als Amiga-Spiel vor. Es geht darum auf einer einsamen Insel die versteckten Teile einer geheimen Rakete zu finden.

Beide Spione suchen zur gleichen Zeit die Insel ab und unterbrechen ihre Tätigkeit nur, um für den anderen eine teuflische Falle zu bauen oder ihm einfach mit einem Knüppel einen neuen Scheitel zu ziehen.

Spy vs. Spy kann alleine gegen den Computer oder mit einem zweiten menschlichen Teilnehmer gespielt werden. Für jeden der beiden Kontrahenten ist eine Hälfte des Bildschirms reserviert, in der sowohl ein Ausblickfenster als auch eine Liste der eingesammelten Gegenstände zu finden ist.

Seit ihrem ersten Auftritt auf dem C-64 haben die beiden Spione nur wenig dazugelernt. Spielprinzip und Musik lehnen sich eng am C-64-Original an, dementsprechend popelig und angestaubt präsentieren sie sich dann auch. Eigentlich hätte man aus den beiden Spionen ein feines

Spiel machen können, doch was hier präsentiert wird, entstammt geradewegs dem Computermittelalter und ist irgendwo zwischen erträglich und ärgerlich anzusiedeln.

(hs)

### AMIGA DOS Blitzlicht

**Name:** Spy vs. Spy  
**Hersteller:** Wicked  
**Quelle:** Fachhandel  
**Preis:** 29,95 DM

**Positiv:**  
– Zwei-Spieler-Option

**Negativ:**  
– grauslicher Sound  
– magere Grafik  
– miese Animation



Der kleine Held, der prügelt munter, die Mutanten die Bunkertreppe runter

## After the War

Nach dem Dritten Weltkrieg wird es duster um unsere ach so heiß geliebte Zivilisation.

Schenkt man der allgemein beliebtesten Endzeitvision Glauben, werden die wenigen, die den Krieg überleben, nichts Besseres zu tun haben, als einander das Fell über die Ohren zu ziehen.

So zieht ein weißblonder Jeansabenteurer durch die Ruinen einer Stadt, auf der Suche nach einem Bunkereingang. Denn während er durch die bedrohliche Landschaft zieht, ist sein Körper unentwegt der gefährlichen Strahlung ausgesetzt, die ihn irgendwann einmal hinraffen wird. Es sei denn, es gelingt ihm, zeitig genug den Bunker zu finden. Doch neben der Strahlung treiben sich noch tollwütige Hunde, brutale Kettenrockers, fiese Killerpunks und amoklaufende Excops in den Trümmern herum, und alle verstellen ihm den Weg in den Bunker. Glücklicherweise kann er unterwegs neben Waffen auch stärkende Getränke und Medikamente gegen die Verseuchung finden.

Vertikal scrollende Prügel-Epen sind seit Double Dragon nichts Neues. After the War ist zwar nicht in der Lage, das Thema originell zu variieren, aber dafür wird es in handfesten Genrebildern interpretiert. Es fehlen weder die Übermutanten am Ende jedes Levels, noch die in Trümmern liegende Freiheitsstatue in der Hintergrundgrafik.

Alles zusammengekommen ist das Spiel zwar solide gemacht, aber nicht gerade originell. Davon einmal abgesehen leidet das Spiel an demselben Übel, das auch alle anderen Vertreter des Genres plagt: Einmal durchgespielt ist schnell die Luft raus und die Spielmotivation sinkt rapide.

(hs)

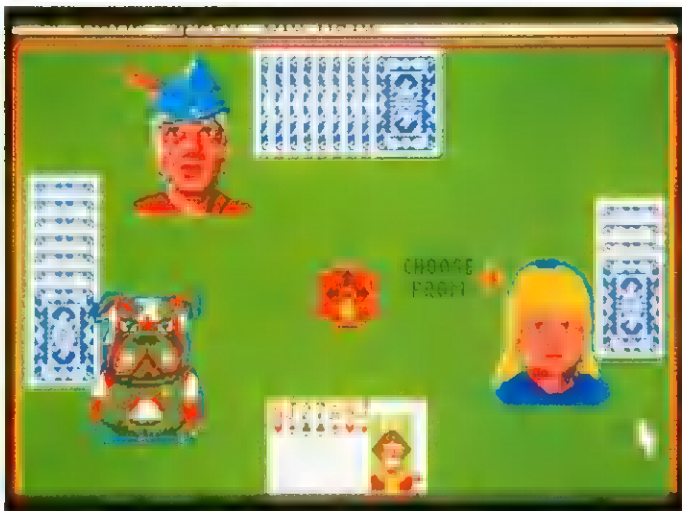
### AMIGA DOS Blitzlicht

**Name:** After the War  
**Hersteller:** Dynamics  
**Quelle:** Fachhandel  
**Preis:** stand bei Redaktionsschluss noch nicht fest

**Positiv:**  
– gute Grafik  
– probate Steuerung

**Negativ:**  
– wenig Neues  
– wird schnell langweilig





Fordern Sie Sir Graham aus King's Quest zu einem Spiel Cribbage

## Hoyle official Book of Games Volume 1

Im frühen 18. Jahrhundert faßte Edmond Hoyle die Grundregeln von Kartenspielen in einem Buch zusammen. Sechs dieser Spiele wurden in diesem Programm vereinigt, und als Gegner kann man zwischen richtigen Profis oder alten Bekannten aus den Sierra-Abenteuern wählen.

**K**arten zu spielen ist immer noch eine der beliebtesten Freizeitbeschäftigungen.

Man hat die Wahl zwischen: Crazy Eights, das etwa unserem Mau-Mau entspricht, Old Maid, unser Schwarzer Peter, Hearts, Gin Rummy und Cribbage, wo zusätzlich zu den Karten auch noch ein Spielbrett auf den Tisch gelegt wird, sowie Klondike Solitaire.

Nach der nun folgenden Gegnerwahl steht dem eigentlichen Spiel nichts mehr im Wege. Die Gesichter der Gegner sind allesamt animiert, und wenn Sie jemanden auswählen, erfahren Sie aus einer Sprechblase etwas über seine Geschichte und Spielstärke. Während des Spiels können Sie jederzeit die Regeln aus einem Menü heraus abfragen.

Hoyle Book of Games ist ohne Zweifel ein Spiel für die gesamte Familie, vom jüngsten bis zum ältesten Mitglied. Die guten Grafiken – der Schwarze Peter ist für die Kleinen sogar animiert – und die vielen unterschiedlichen Gegner verleihen diesem Spiel lange Zeit Spielspaß.

(Robert Marz/hs)

### AMIGA DOS Blitzlicht

**Name:** Hoyle official Book of Games

**Hersteller:** Sierra

**Quelle:** Fachhandel

**Preis:** stand bei Redaktions-schluß noch nicht fest

#### Positiv:

- unterschiedliche Spiele mit verschiedenen Stärken
- gute Grafiken
- Regeln jederzeit abrufbar
- Save-Settings-Option

#### Negativ:

- Sound nervend, aber abstellbar
- komplett in Englisch



Mit dem Hubschrauber im Kampf gegen das Böse auf der Welt

## FIRE!

Drogen, Terrorismus und Kriege sind allesamt Probleme, die den Frieden der Welt bedrohen. Mit dem Kampfhubschrauber Fire und dem Kriegsschiff USS New Deal als Basis können Sie diesen Problemen ein Ende bereiten. Also die Bordkanonen vorglühen und auf in den Kampf.

**E**s gilt, fünf gefährliche Aufträge zu lösen, von denen sich jeder in zwei Teile aufteilt. Da wären zum Beispiel der Drogenhandel in Kolumbien oder die Kämpfe im Libanon.

In Grünland hat eine Verbrecherorganisation ihr Hauptquartier aufgeschlagen und U-Boote zwischen Wäldern versteckt. Schließlich müssen Sie noch den Völkermord in Südostasien beenden und eine kriegerische Nation bändigen.

In den einzelnen Levels ist der Spielablauf immer gleich: Sie fliegen mit dem Hubschrauber durch eine horizontal in beide Richtungen scrollende Landschaft. Im oberen Bereich fliegen Hubschrauber, Flugzeuge und Raketen, während sich unten die eigentlichen Ziele befinden. Um ein Level zu beenden, muß man alles abschießen, was spätestens ab Level vier ziemlich schwer wird, da auch der Treibstoff ausgehen kann. Ab und zu verliert ein abgeschossenes Flugzeug einen Doppelschuß für den Hubschrauber oder eine Tonne mit zusätzlichem Fuel, die allerdings relativ schnell auf den Boden fallen. Das sind die einzigen Extrawaffen in diesem Spiel. Fire! ist ein Ballerspiel ohne besondere Extras oder spit-

zenmäßige Grafik. Daher ist es wohl eher in den mittleren Qualitätsbereichen anzusiedeln. Es ist aber leicht spielbar, und durch die Continue-Option motiviert es den Spieler ungemein, zumindest bis das Spiel geschafft ist.

(Robert Marz/hs)

### AMIGA DOS Blitzlicht

**Name:** Fire!

**Hersteller:** New Deal

**Preis:** 89,- DM

#### Positiv:

- verschiedene Aufträge
- Continue-Option

#### Negativ:

- wenig Extras
- kriegerische Handlung







Eins, zwei, drei, im Sauseschritt, rast Space Harrier über Amiga-Monitore

## SPACE HARRIER II

### Return to the Fantasyzone

Die Sonnenbrille vor die Augen geschoben, den tragbaren Laser unter dem Arm und die Jetboots an den Füßen, dürfen nun auch Amiga-Benutzer den zweiten Aufguß des Space-Harrier-Themas genießen.

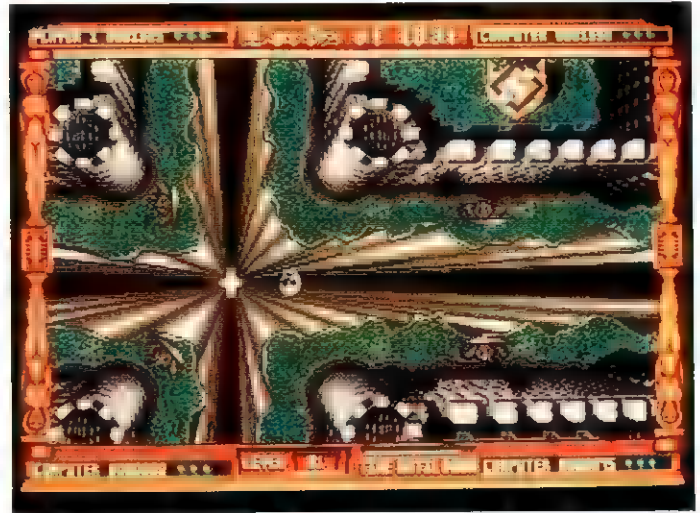
**E**s gilt die verschiedenen Welten der Fantasyzone von eingedrungenen Aliens zu befreien. Während der Held die einzelnen Welten durchquert, versuchen die Biester ihn zu stoppen, indem sie sich mit vereinter Kraft, Welle für Welle auf ihn stürzen. Sein einziger Schutz ist die schwere, tragbare Laserkanone, die er bei sich hat. Die Zonen, durch die der Held muß, werden als dreidimensionale Szene dargestellt. Der Boden ist in ein Schachbrettmuster aufgeteilt und wird so animiert, daß der Spieler den Eindruck hat, über eine Ebene hinwegzuschweben. Unterdessen erscheinen die Ungeheuer aus der Tiefe des Raumes, oder aber im Rücken des Helden. Aus allen Rohren feuernd ziehen sie eine festgefügte Bahn durch das Szenario und verschwinden dann wieder, sofern sie bis dahin nicht schon abgeschossen wurden.

Space Harrier II ist bunt, laut und schnell. Leider ändern die Gegner ihre Angriffstaktiken im Laufe des Spiels kaum und der steigende Schwierig-

keitsgrad kommt im Grunde nur durch verstärkte Alienattaken zustande.

(hs)

AMIGA DOS Blitzlicht	
<b>Name:</b>	Space Harrier II
<b>Hersteller:</b>	Elite
<b>Quelle:</b>	Fachhandel
<b>Preis:</b>	64,95 DM
<b>Positiv:</b>	- gute Musik
<b>Negativ:</b>	- wenig abwechslungsreich - Gegner ■■ schnell



Breakout war einer der Ahnherren dieses Spiels, nur hier werden keine Steine zerschlagen, sondern Burgmauern

## Lords of War

Altbekannte Spielideen erleben immer wieder Neuauflagen. Und wenn das Spielkonzept so abgedroschen ist, daß es keinen Hund mehr hinter dem Ofen hervorlockt, dann wird ■ eben mit einer anderen Spielidee gekreuzt.

**E**benso wurde mit Lords of War verfahren. Breakout und Arkonoid sind die ursprünglichen Paten, denn es geht darum, einen Ball mit einem Schläger zu treffen.

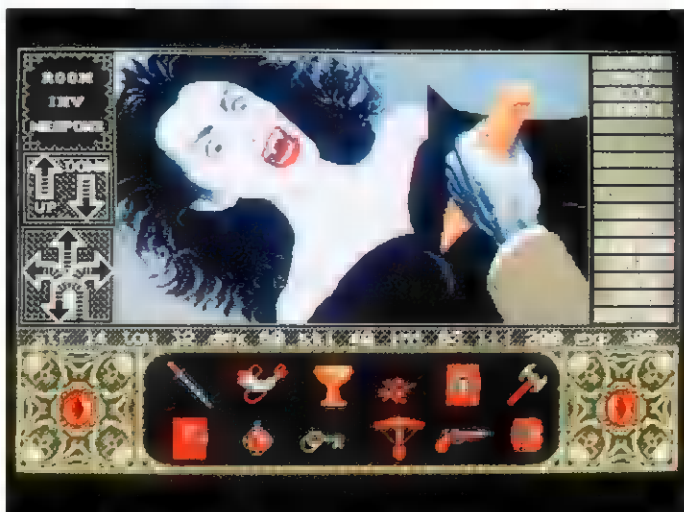
Vier Burgen stehen an den vier Eckpunkten eines Rechtecks. In dem Raum zwischen den Mauern springt ein Ball hin und her. Vor jeder Burg ist ein Schild angebracht, der rund um den Wall bewegt werden kann. Wahlweise kann nun jede der vier Burgen durch einen menschlichen Spieler oder aber durch den Computer gegen den umherfliegenden Ball verteidigt werden. Dazu muß der Schild so vor der Mauer verschoben werden, daß der umherfliegende Ball durch den Schild umgelenkt wird. Trifft der Ball, der übrigens wie ein Totenschädel aussieht, die Mauer, schlägt er aus dieser ein kleines Stück heraus. Im Innenhof jeder Burg liegt ein Wappen, das durch die umgebenden Mauern geschützt wird. Zerschlägt der Ball die Mauer, gelangt in den Innenhof und trifft dort das Wappen, ist das Spiel für den Burgherren verloren.

Insbesondere daß vier Spieler gleichzeitig spielen können,

verleiht diesem Spiel Reiz. Zaubersprüche und andere Extras machen aus Lord of War ein passables Geschicklichkeitsspiel, das auch nach längerem Spielen noch viel Spaß macht.

(hs)

AMIGA DOS Blitzlicht	
<b>Name:</b>	Lords of War
<b>Hersteller:</b>	Digital Concepts
<b>Quelle:</b>	Fachhandel
<b>Preis:</b>	stand bei Redaktions-schluß noch nicht fest
<b>Positiv:</b>	- Vier-Spieler-Option - Zaubersprüche
<b>Negativ:</b>	- Grafik wenig abwechslungsreich



Allerlei Horrorgetier erwartet Sie in Elvira, Mistress of the Dark



Hägar – Jetzt auch auf dem Amiga schrecklich aktiv!

## Elvira, Mistress of the Dark

Bisher durfte die attraktive Elvira sich nur im Firmensignet von Horrorsoft rekeln, jetzt hat sie ihr eigenes Horror-Rollenspiel, das geneigten Spielern mehr als einen Schauer über den Rücken jagen wird. Elvira hat das mittelalterliche Schloß einer ihrer Vorfahren geerbt, natürlich komplett mit allem möglichen unheiligen Monstergezücht. Was bisher an Grafik und Sound zu sehen und zu hören war, war sehr vielversprechend.

## Hägar, der Schreckliche

Hägar, der beliebte Comic-Wikinger, erobert den Amiga. Auch in seinem Computerspiel darf Hägar seiner liebsten Beschäftigung nachge-

# Demnächst auf Ihrem Computer

**Viele alte und neue Bekannte aus anderen Medien, wie zum Beispiel Comic und Film, erleben ihr Debut als Computerspielhelden. Es erwarten Sie Hägar, einer der schrecklichsten und beliebtesten Wikinger der Welt, und Elvira, die Lady der Finsternis und bekannte Komödien-Horror-Sexbombe amerikanischer Mitternachtsshow.**

hen, ferne Länder zu plündern. Allerdings darf er nicht vergessen, seiner lieben Helga etwas mitzubringen. Und da es sich mit blanken Händen und leerem Magen schlecht kämpft, gilt es, sich zunächst mit Rüstzeug und Leckereien zu versehen.

## Pipe Mania

Mit diesem Titel steht uns ein feines Geschicklichkeitsspiel ins Haus, in dem es um Pipelines geht. Der Spieler muß aus verschiedenen geformten Teilstücken eine durchgehende Rohrleitung basteln. Nach

einer gewissen Zeitspanne dreht jemand den Hahn der Leitung auf, und dann wird sich erweisen, wie gut das Rohr zusammengesetzt ist. Jede undichte Stelle rächt sich durch Punktabzug am endgültigen Score.

## Venus

Ein neues, etwas anderes Ballespiel wird Venus werden, von dem wir schon eine Demoversion spielen konnten. Ein insektenartiges Kunstwesen muß dabei versuchen, allerlei boshaften Monstern das Lebenslichtlein auszupusten. Neben Extrawaffen, guter Grafik und sauberem Scrolling erwarten den Spieler auch Zonen mit verkehrter Gravitation. Dann läuft Ihr Käferandroid auf dem Kopf stehend am oberen Bildschirmrand entlang.

(hs)



Pipelines zu bauen, ist nicht gerade leicht, besonders wenn mitten in der Arbeit die Schleusen geöffnet werden



Mal normal, mal auf dem Kopf, Venus präsentiert sich in opulenter Farbenpracht



## Impressum

### Herausgeber

Christian Widuch

### Chefredaktion

Markus Matejka (mm)

### Redaktion

Jürgen Borngießer (jb), Bernhard Rinke (br),  
Heinrich Stiller (hs), Joachim Freiburg (jf),  
Martin Schlöter (ms), Vera Brinkmann (vb)

### Freie Autoren dieser Ausgabe

Edgar Meyziz, Bernd Rudolf, Robert Marz,  
Jürgen Seibel, Garry Glendown, Ulf Petersen,  
Michael Anton, Andreas Polk, Sebastian Ritter,  
Stefan Hochmuth, Ralf Zuber, Willi Bäcker,  
Monika Schmid

### Redaktionsassistent

Anke Kerstan (ke), Susanne Eska (es)

### Koordination

Stefan Ritter

### Produktionsleitung

Gerd Köberich

### Bereichsleitung

Claudia Ebbrecht (Fotosatz/Lektorat),  
Margarete Schenk, Helmut Skoupy  
(Montage/Reprografie)

### Layout

Michael Grebenstein

### Fotografie

Christian Heckmann

### Fotosatz

Gabriela Joseph

### Lektorat

Susanne Lessinger

### Montage/Reprografie

Manuela Eska, Peter Gajewski

### Werbegestaltung

Mohamed Hawa, Petra Küch

### Anzeigenverkaufsleitung

Wolfgang Brill

### Anzeigenverkauf

DMV-Verlagsbüro München  
Zaunkönigweg 2c, 8000 München 82  
Telefon (089) 4 39 10 87, Telefax 0 89/4 39 10 80  
Leitung: Britta Fiebig  
Anzeigenverkauf: Monika Schöbel, Jens Dhein,  
Peter Schätzle

### Anzeigenpreise

Es gilt die Anzeigenpreislise Nr. 1 vom 01.01.1990

### Anschrift Verlag/Redaktion:

DMV Daten & Medien-Verlag  
Widuch GmbH & Co. KG  
Fuldaer Straße 6  
3440 Eschwege, Telefon (0 56 51) 8 09-0,  
Telefax (0 56 51) 8 09-333

### Vertrieb

Verlagsunion Erich Pabel-Arthur Moewig KG (VPM),  
Friedrich-Bergius-Straße 20, 6200 Wiesbaden

### Druck

Druckerei Jungfer, 3420 Herzberg

### Bezugspreise

»AMIGA DOS« erscheint monatlich.  
Einzelpreis DM 6,50/sfr. 6,50/6S 52,-

### Abonnementpreise

Die Preise verstehen sich grundsätzlich einschließlich  
Porto und Verpackung.

### Inland:

12 Ausgaben: DM 70,-  
6 Ausgaben: DM 35,-

### Europäisches Ausland:

12 Ausgaben: DM 100,-  
6 Ausgaben: DM 50,-

### Außereuropäisches Ausland:

12 Ausgaben: DM 120,-  
6 Ausgaben: DM 60,-

### Bankverbindungen:

Postscheck Frankfurt/M: Kto.-Nr.: 23043-608  
Raiffeisenbank Eschwege:  
BLZ: 522 603 85, Kto.-Nr.: 245 7008

Die Abonnementbestellung kann innerhalb einer  
Woche nach Auftrag beim DMV-Verlag, Postfach 250,  
3440 Eschwege, schriftlich widerrufen werden. Zur  
Wahrung der Frist reicht der Poststempel. Das Abonne-  
ment verlängert sich automatisch um 6 bzw. 12 Monate,  
wenn es nicht mindestens 6 Wochen vor Ablauf beim  
Verlag schriftlich gekündigt wird.

Für unverlangt eingesandte Manuskripte und Datenträ-  
ger sowie Fotos übernimmt der Verlag keine Haftung.  
Die Zustimmung zum Abdruck wird vorausgesetzt.

Eine Haftung für die Richtigkeit der Veröffentlichungen  
kann trotz sorgfältiger Prüfung durch die Redaktion vom  
Herausgeber nicht übernommen werden. Die geltenden  
gesetzlichen und postalischen Bestimmungen sind zu  
beachten.

Die gewerbliche Nutzung, insbesondere der Programme,  
Schaltpläne und gedruckten Schaltungen, ist nur mit  
schriftlicher Genehmigung des Herausgebers zulässig.  
Das Urheberrecht für veröffentlichte Manuskripte liegt  
ausschließlich beim Verlag. Nachdruck sowie Vervielf-  
ältigung oder sonstige Verwertung von Texten nur mit  
schriftlicher Genehmigung des Verlages.

Namentlich gekennzeichnete Fremdbeiträge geben nicht  
in jedem Fall die Meinung der Redaktion wieder.

## Die Inserenten

ABAG .....	69	HK-Computer .....	19
ABC-Soft .....	17	Kramer .....	121
AFM Computer .....	120	Mac-Soft .....	121
Alpha 2000 .....	133	Musik- und Grafiksoftware Shop .....	121
Alpha Soft .....	120	MVC .....	82
A.P.S Elektronik .....	120	Newcom .....	103
Auriga .....	87	Nordsoft .....	23
B + C EDV-Systeme .....	35	Nürnberger PD-Studio .....	71
B + S Computer .....	121	Ossowski .....	147
CIK-Computertechnik .....	120	P.V.-Computershop .....	121
CLS Computerladen .....	133	Ptak PD-Versand .....	121
Compi Mate .....	35	Peksoft .....	135
CompuCamp .....	Postkarte	Reis Ware .....	27
Computer-Express .....	113	Roßmüller Handshake .....	67
Computing .....	41	Schewe .....	103
Compy-Shop .....	41	Scholle .....	23
CPS Computertechnik .....	43	Sky Ware .....	87
CSV Riegert .....	69	Strauß Elektronik .....	11
Data 2000 .....	101	TS-Datensysteme .....	139
Diezemann .....	53	Urlbauer .....	120
DMV .....	21,78,79,148	VESALIA Computer .....	47
Dombrowski .....	121	Wegener Technischer Vertrieb .....	57
Donau-Soft .....	82,121	WENNGATZ .....	120
Fester .....	120	Wolf-Computertechnik .....	61
Gold Disk .....	2		
Gussenbauer Software .....	121		



## Im nächsten Heft

■ ■ ■ **Mit BTX geht's fix!** Wer ein Postmodem für BTX hat, der braucht nur noch die passende Software und ein entsprechendes Kabel. Wir haben uns in BTX umgesehen.



Bild 1. Digi-Tiger gegen den Rest der Welt: Wir haben uns den handlichen Digi-Tiger einmal für Sie angesehen. Was er alles zu leisten vermag, lesen Sie in der nächsten Ausgabe

■ ■ ■ **386er Power:** Von der Firma Roßmüller gibt es eine Neuheit für PC-Karten-Besitzer: eine Erweiterungskarte mit einem 386SX-Prozessor. Ob der Amiga damit zum Super-PC wird, erfahren Sie im nächsten Heft.

■ ■ ■ **Amiga und Drucker** – ein nicht immer unproblematisches Thema. Wir haben den Star LC24-15 einem Test unterzogen. Das Ergebnis finden Sie in AMIGA DOS 6/90.

■ ■ ■ **Das haben Sie schon immer gesucht:** Jede Menge neuer Tips & Tricks zu Amiga-BASIC, CLI, Assembler, Modula2, C, GFA-BASIC warten auf Sie.

■ ■ ■ **Topaktuell...** Die neuesten Spiele auf dem Softwaremarkt haben wir für Sie gesichtet. Außerdem wieder jede Menge Tips zu den beliebtesten Spielen auf dem Amiga in unserer Helpline.

■ ■ ■ **Und natürlich...** eine geballte Ladung Listings, Tests und Infos.

■ ■ ■ **DFÜ** – Begleiten Sie uns auf einem ausgedehnten Streifzug durch die Welt der Datenfernübertragung. Ob Modems, Koppler oder Übertragungsprotokolle, AMIGA DOS hat alles Wissenswerte über Hard- und Software zu diesem Thema für Sie zusammengestellt. Außerdem: eine Liste von interessanten Mailboxen.



Bild 2. Die 386SX-Karte – Lohnt sich der Einbau?

Die nächste  
**AMIGA**  
DOS  
finden Sie ab

**09. Mai '90**

bei Ihrem Zeitschriftenhändler



Bild 3. Venus: Kampf den Käfern ist der Inhalt dieses sehenswerten Shoot'em Ups. Gute Grafik und reichlich Action – mehr in der nächsten AMIGA DOS



# STEFAN OSSOWSKI'S SCHATZTRUHE

- 1 **Haushaltsbuch** bis zu 25 Konten, flexibel, leicht bedienbar, mit umfangreicher **deutschsprachiger** Dokumentation DM 8,-
- 3 **MountainCAD** professionelles CAD-Programm, **deutsche Anleitung** DM 8,-
- 4 **Spiele I, II, III** 10 erstklassige PD-Spiele aus allen Bereichen wie Action, Geschicklichkeit, Strategie (3 Disketten) DM 24,-
- 5 **Anti-Virus 8** Programme gegen alle Viren DM 8,-
- 6 **Text** hochwertige **deutsche** Textverarbeitung DM 8,-
- 7 **Utility-Disk** 25 nützliche Utilities aus allen Bereichen DM 8,-
- 9 **Sonix-Paket** Original-Sonix-Player + 4 weitere Disketten mit phantastischer Sonix-Musik. **Top-Hit!** DM 40,-
- 10 **Business 3** Disketten: Tabellenkalkulation, relat. Datenbank, sehr gute Textverb. Vers. engl. DM 24,-  
Vers. deutsch DM 70,-
- 13 **Paranoid** sensationelles Breakout-Spiel DM 8,-
- 14 **Buchhaltung** erstes **deutsches** PD-Buchhaltungsprogr. DM 8,-
- 16 **AMIGA-PAINT** sehr gutes **deutsches** Malprogramm DM 8,-
- 17 **Videodatei** bringt Ordnung in Ihre Videodatei, **deutsch** DM 8,-
- 18 **Fußballmanager** bei diesem Spiel können Sie Ihre Fähigkeiten als Manager eines Fußballclubs testen, **deutsch** DM 8,-
- 20 **Giroman** komfortables **deutsches** Programm, mit dem Sie Ihr Girokonto einfach verwalten können DM 8,-
- 22 **Kampf um Eriador, V.2.0** taktisches Strategiespiel für 2 Personen mit sehr guter Grafik und Sound, **deutsch** DM 8,-
- 26 **Risiko** die Amiga-Umsetz. d. bek. Brettspiels, **deutsch** DM 8,-
- 27 **Ray-Tracing-Construction-Set, V.2.0** phantastisches Programm zur Berechnung von Licht und Schatten - siehe Test Amiga 1/88 - komplett auf 3 Disketten mit **deutscher Anleitung** DM 24,-
- 28 **Wizard of Sound** ein phantastisches Musikprogramm zur Erstellung eigener Lieder, mit **deutscher Anleitung** (2 Disks) DM 10,-
- 29 **Broker** ein sehr gutes **deutsches** Börsenspiel DM 8,-
- 30 **Quickmenü** erst. Sie sich Ihre eig. Workbench i. **deutsch** DM 8,-
- 31 **Blizzard** phantastisches Ballerspiel m. sehr guter Animat. DM 8,-
- 32 **DSort** **deutsches** Diskettenkatalogisierungsprogramm DM 8,-
- 33 **Pascal** ein komplettes Pascal-Paket (3 Disketten) mit Compiler, **deutscher Anleitung** u. einem s. gut. deutschen Editor DM 24,-
- 34 **DiskKey** Diskettenmonitor mit **deutscher Anleitung** DM 8,-
- 35 **Peters Quest** Geschicklichkeitsspiel mit lustiger Handlung und **deutscher Anleitung** DM 8,-
- 36 **Spiele** auf dieser Diskette sind 3 Tetris-ähnli. Spiele enth. DM 8,-
- 37 **MRBackup** Festplattensicherungsprogramm mit **deutscher Anleitung** DM 8,-
- 38 **Universal-Datei** **deutsches** Datei-Verwaltungsprogr. DM 8,-
- 39 **Assembler** ein komplettes Entwicklungssystem für Maschinensprache in **deutsch!** DM 8,-
- 40 **Bibel-Quiz** lehrreich und unterhaltsam DM 8,-
- 42 **BootMaster** Mit diesem Programm können Sie individuelle Bootblöcke mit Lauftext und Sternenhintergrund erstellen. DM 10,-
- 43 **Banner II** Dieses Programm ermöglicht es Ihnen, komplette Banner mit Ihrem Drucker zu erstellen. Leicht bedienbar! DM 8,-
- 44 **Boulder V1.3** Boulder ist ein **sehr schnelles Geschicklichkeitsspiel**, das an den C64-Klassiker **Boulder-Dash** angelehnt ist. Ohne Sound! DM 8,-
- 45 **Label-PAINT** **deutsches** Etikettendruckprogr. mit Grafik! DM 8,-
- 46 **Roll On** friedliches Geschicklichkeitsspiel mit Leveleditor, Ein Spiel mit langanhaltender Motivation. Super! DM 8,-
- 47 **Paccy** der alte Spielhallenklassiker lebt wieder auf DM 10,-
- 48 **Pente** Ein intelligentes "5-Gewint"-Spiel. **Deutsch!** DM 10,-
- 49 **Tumbler Street** Glücksspiel, bekannt durch **Salvatore** von **RTL-Plus!** Benötigt 1 MB Speicher! DM 8,-
- 101 **RIM-5 = Relationale Datenbank**  
Außerst leistungsfähig, sowohl für den privaten als auch für den geschäftlichen Bereich geeignet. **Mit deutscher Anleitung und ausführlichem Einführungskurs.** DM 30,-
- 102 **AnalytiCalc = Tabellenkalkulation**  
**Leistungsstark mit deutscher Anleitung.** Ein unverzichtbares Hilfsmittel für Kalkulationsaufgaben jeder Art (**Bericht Amiga 12/89**). DM 30,-
- 103 **DEA Arithmetica = Die Göttin Arithmetica**  
Besonders für Schüler, Studenten und Lehrer geeignet. Ableitungen, **Kurvendiskussion** und Skizzieren von Funktionsgraphen problemlos möglich. **Deutsch!** DM 30,-

- 104 **Haushaltsbuch Version 2.1**  
Komplettlösung zur Verwaltung Ihrer privaten Finanzen! Leicht bedienbar, auch für Computer-Neulinge geeignet. Unterstützung durch Demo-Dateien, frei von buchhalterischen Fachausdrücken. Funktionsüberblick: verschiedene Dateien, mehrere Kontenlisten, bis zu 10 Bilanzen, 40 freidefinierbare Konten, Suchroutinen, doppelte Buchführung, Filterfunktionen, Jahresübertrag, Mausunterstützung... **Deutsche Dokumentation im Ringbuch!** 1 MB Speicher erforderlich! DM 98,-
- 105 **Xytronic II**  
Intergalaktische Handelssimulation mit guter Grafik und gutem Sound sowie sehr hoher Motivation. **Natürlich in Deutsch!** DM 29,-
- 106 **Data-Manager**  
Eine sehr leicht zu bedienende Dateiverwaltung, mit der Sie Adressen, Mitgliederlisten etc. perfekt verwalten können. **Deutsches Programm mit deutscher Anleitung!** DM 15,-
- 108 **Bundesliga 2000**  
Programm zur Bundesligaergebnisverwaltung und Tabellenberechnung. Verwaltung beliebig vieler Ligen möglich. Historie seit 1985. **Deutsch!** DM 15,-
- 109 **Money Player Deluxe - Geldspielgerät**  
Sie fühlen sich wie in der Spielhalle! Start- und Risikoautomatik, **Sonderspiele** viele Extras und toller Spielspaß. **Palaauflösung, Maussteuerung!** DM 39,-
- 110 **Wizard of Sound 2.0 - Musikprogramm**  
WoS 2.0 ist ein sehr gutes Musikprogramm mit 61 Instrumenten, kompletter Notation, Player-Programm, Instant-Replay-Modus, einfachste Noteneingabe, ... Auch für Musik-Laien geeignet! **Auf 2 Disketten mit deutscher Dokumentation!** DM 35,-
- 111 **Broker 1.2 - Das Börsenprogramm**  
Sehr **realistische Börsensimulation**, mit bis zu 4 Spielern spielbar. Broker 1.2 ist unterhaltsam, spannend und lehrreich. **Deutsch!** 1 MB Speicher! DM 29,-
- 112 **Vokabel-Trainer-Englisch 1.0**  
Leicht zu bedienendes, leistungsfähiges Lernprogramm. Der Wortschatz ist individuell erweiterbar. Damit macht das Lernen Spaß! **Deutsch!** DM 15,-
- 114 **BootMenü**  
Sie können im Disketten-Bootblock ein Menü einrichten, über das Sie beim Bootvorgang per Maus 10 verschiedene Programme aufrufen können! **Mit deutscher Anleitung!** DM 19,-
- 115 **CBB - CopyBootBlock Installierungsprogramm**  
Mit CBB können Sie in Ihrem Bootblock ein Kopierprogramm installieren, das sie bei jedem Bootvorgang per Mausklick starten können. 2 Laufwerke oder 1 MB. DM 19,-
- 116 **Danger Castle**  
Ein Hüpf- und Rennspiel für 1 oder 2 Personen mit **Wahnsinnsgrafik, Stereound** und 20 verschiedenen Levels! **Deutsche Anleitung!** DM 39,-
- 117 **SpeedRunner**  
Eine Loadrunner-Variante mit Level-Editor, 100 Levels, **Super-Grafik und Klasse-Sound.** In Assembler geschrieben und **Palaauflösung!** **Deutsche Anleitung!** DM 39,-
- 119 **Einkommensteuer '89**  
Mit diesem Programm können Sie sehr einfach Ihre Lohn- und Einkommensteuer berechnen. Deckt 99 % aller Fälle ab. Es werden viele Sonderfälle behandelt. **Deutsch!** DM 79,-
- 120 **Chemie auf dem Amiga**  
Ein didaktisch hervorragend aufgebautes Lernprogramm für die Klassen 7-11. Einfachste Bedienung, viele erläuternde Grafiken und ein breites Themenspektrum zeichnen dieses Programm aus. DM 49,-
- 121 **Tape It!**  
Ein professionelles Programm zur Verwaltung Ihrer Musiksammlung. Vielfältige Listen können erstellt und Kassettenhüllen bedruckt werden. **Deutsch!** DM 19,-
- 122 **SchreibM 2.0 - Schreibmaschinentrainer**  
Spielend lernen Sie Ihre Tastatur in den Griff zu bekommen. DM 19,-
- 123 **K.A.L. - Verwaltung**  
Einfache Kunden-, Artikel- und Lieferantenverwaltung DM 19,-
- 124 **SGM - Statistik-Grafik-Manager**  
Auf einfache Art und Weise können Sie mit **SGM** Statistik- und Präsentationsgrafiken erstellen: Balken-, Torten-, Tendenz-, Flächengrafiken. Diese Grafiken können ausgedruckt oder im IFF-Format weiterbearbeitet werden. **Deutsch!** DM 49,-
- 125 **Lotto V.1.01**  
Auf dieses **deutsche** Programm haben Lotto-Fans schon lange gewartet. Mit **Lotto V.1.01** können Sie Ergebnisse verwalten, Tipps erstellen, Statistik betreiben, u.v.m. ... Mit VEW-System! DM 19,-
- 126 **AmigaDraw V.1.2**  
Ein professionelles **deutsches** Mal- und Zeichenprogramm DM 19,-
- 127 **Hawk Inc.**  
Ihre Aufgabe ist es, einen schweren Fall von Wirtschaftskriminalität aufzudecken! Dringen Sie in den Zentralcomputer ein und lösen Sie den Fall! **Ein realistisches und spannendes Spiel!** DM 19,-
- 128 **Aktien-Depot-Manager**  
Eine Aktien-Verwaltung par Excellence! Mit grafischer Chart-Darstellung, Depot-Management, vielen Optionen, chartanalytische Kaufempfehlungen, etc. ... **Mit umfangreicher deutscher Anleitung!** DM 49,-
- 201 **CLI-Help-Deluxe Software für Einsteiger!**  
Mit dieser Tutorial-Diskette lernen Sie das CLI kennen und beherrschen! Speziell auf die Bedürfnisse von **Anfängern und Einsteigern** ist diese hochinformativ Diskette abgestimmt. Mit dieser Diskette gehören Probleme mit Amiga-DOS der Vergangenheit an! DM 19,-

\* TAIFUN \* TAIFUN \* TAIFUN \* TAIFUN \* TAIFUN \* TAIFUN \* TAIFUN \* TAIFUN \* TAIFUN \* TAIFUN  
Die deutsche Public-Domain-Serie aus dem Hause Ossowski! Wußten Sie schon, daß beim PD-Versand Stefan Ossowski alle drei Monate 10 neue TAIFUN-Disketten mit den interessantesten Neuerscheinungen des PD-Marktes vorgestellt werden? Bevorzugt präsentieren wir Ihnen die neuesten deutschen Programme. Am 15. Februar erschienen die neuen TAIFUN-Disketten Nr. 121 bis 130.  
Schnupperpreis: DM 53,- V-Scheck, DM 57,- Nachnahme

**Versandkosten** Inland: DM 3,- V-Scheck DM 7,- Nachn.  
**(Porto/Verpackung):** Ausland: DM 6,- V-Scheck DM 15,- Nachn.

## ABO-SERVICE

Bei uns erhalten Sie fast jede PD-Serie auch im günstigen Abonnement! Auf unsere Staffelpreise gewähren wir außerdem einen **10%igen ABO-Rabatt!** Die Fish-Serie ist z. B. schon bis Nr. 310 lieferbar! Rufen Sie uns doch einfach an oder schreiben Sie uns, wenn Sie an weiteren Informationen zu unserem ABO-Service interessiert sind!

Zuverlässigkeit  
+ Schnelligkeit  
+ Service  
= PD-Versand Stefan Ossowski  
**Testen Sie uns!**

Wir führen alle bekannten PD-Serien wie Fish, Panorama, Faug Amicus, Auge, Taifun, Chiron, RPD, Kickstart, Slideshows, TBAG, Franz ... Viele Programme deutschsprachig.

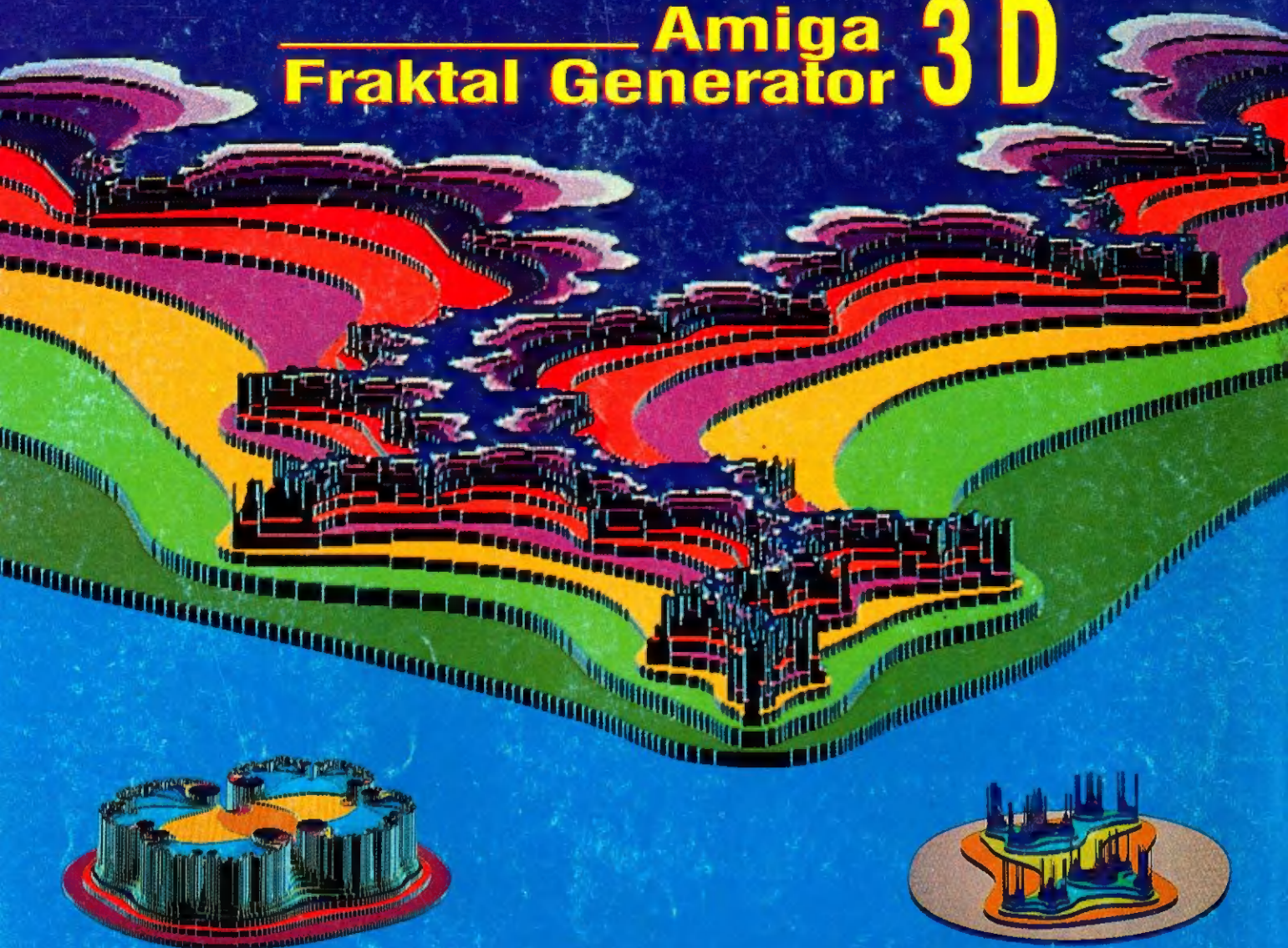
# 5,-DM

kosten unsere aktuellen 2 Katalogdisketten. Mit deutschem Inhaltsverzeichnis unseres gesamten PD-Angebots von weit über 1500 Disketten. Ab DM 4,50. Gegen Verrechnungsscheck oder in Briefmarken anfordern.

**Stefan Ossowski - Entwicklung und Vertrieb von Software und Public Domain**  
Veronikastraße 33, D-4300 Essen 1, Tel./Btx: 0201/788778



# Amiga 3D Fraktal Generator



## Supergrafik im Sekundentakt

Vergessen Sie alles, was Sie bisher über Fraktalgrafik-Programme gehört haben  
– die unendliche Weite phantastischer Bilder erschließt sich nur über ein  
superschnelles Programm: **Fraktal Generator 3D**

### High-Speed

Nur noch 7 Sekunden für das Urbild!

### Super-Parallel-Projektion

Frei wählbarer horizontaler Blickwinkel mit 360 Grad:  
Betrachten Sie das "Fraktalobjekt" von allen Seiten  
Stufenloser vertikaler Blickwinkel:  
Wahlweise Sicht von oben und unten, schräg oder in der Totalen

### Speicherung im IFF-Standard

Einladen der Fraktal-Bilder in Mal- und Zeichenprogramme:  
Verwendung als Hintergrund, Motiv oder Vorlage

### Voller Bedienungskomfort

Auswahl komplett mit Pull-Down-Menüs  
Wahlweise Maus- oder Tastensteuerung

### Phantastische Farbmöglichkeiten

32 Farben im Low-Resolution-PAL-Modus  
Eigenes Farbrequester mit stufenloser Schiebereglung

### Mehrere separate Bildspeicher

Bis zu vier Bilder gleichzeitig abrufbar  
Separate Farbzuordnung und Animationsmöglichkeit

### Farb-Animationen

Phantastische Effekte durch Amiga-Color-Cycling

## Amiga 3D Fraktal Generator

3 1/2"-Disk. Best.-Nr. 2901

**69,- DM** (unverbindliche Preisempfehlung)

Wenn Sie über den DMV-Bestellservice bestellen, gilt folgendes:

#### Inland:

Einzelpreis 69,- DM

zzgl. Versandkosten 4,- DM

Endpreis 73,- DM

#### Ausland:

Einzelpreis 69,- DM

zzgl. Versandkosten 6,- DM

Endpreis 75,- DM

– Bitte benutzen Sie die Bestellkarte im Heft. –

**DMV-Verlag · Postfach 250 · 3440 Eschwege**

